请注意赛普拉斯已正式并入英飞凌科技公司。
此封面页之后的文件标注有“赛普拉斯”的文件即该产品为此公司最初开发的。请注意作为英飞凌产品组合的部分，英飞凌将继续为新的及现有客户提供该产品。

文件内容的连续性
事实是英飞凌提供如下产品作为英飞凌产品组合的部分不会带来对于此文件的任何变更。未来的变更将在恰当的时候发生，且任何变更将在历史页面记录。

订购零件编号的连续性
英飞凌继续支持现有零件编号的使用。下单时请继续使用数据表中的订购零件编号。

32 位微处理器

FM0+ 家族 外设手册

Doc. No. 002-11330 Rev. **

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone (USA): 800.858.1810
Phone (Intnl): 408.943.2600
http://www.cypress.com

赛普拉斯半导体公司，2013-2016 年。本文件是赛普拉斯半导体公司及其子公司，包括 Spansion LLC（“赛普拉斯”）的财产。本文件，包括其包含或引用的任何软件或固件（“软件”），根据全球范围内的知识产权法律以及美国与其他国家签署条约由赛普拉斯所有。除非在本款中另有明确规定，赛普拉斯保留在该等法律和条约下的所有权利，且未就其专利、版权、商标或其他知识产权授予任何许可。如果软件并不附随有一份许可协议且贵方未以其他方式与赛普拉斯签署关于使用软件的书面协议，赛普拉斯特此授予贵方属人性质的、非独家且不可转让的如下许可（无再许可权）（1）在赛普拉斯特软件著作权项下的下列许可权（一）对以源代码形式提供的软件，仅出于在赛普拉斯硬件产品上使用之目的且仅在贵方集团内部修改和复制软件，和（二）仅限于在有关赛普拉斯硬件产品上使用之目的将软件以二进制代码形式的向外部最终用户提供（无论直接提供或通过经销商和分销商间接提供），和（2）在被软件（由赛普拉斯公司提供，且未经修改）侵犯的赛普拉斯专利的权利主张项下，仅出于在赛普拉斯硬件产品上使用之目的制造、使用、提供和进口软件的许可。禁止对软件的任何其他使用、复制、修改、翻译或汇编。

在适用法律允许的限度内，赛普拉斯未对本文件或任何软件作出任何明示或暗示的担保，包括但不限于关于适销性和特定用途的默示保证。赛普拉斯保留更改本文件的权利，届时将不另行通知。在适用法律允许的限度内，赛普拉斯不对因应用或使用本文件所述任何产品或电路引起的任何后果负责。本文件，包括任何样本设计信息或程序代码信息，仅为供参考之目的提供。文件使用人应负责正确设计、计划和测试信息应用和由此生产的任何产品的功能和安全性。赛普拉斯产品不应被设计为、设定为或授权用作武器操作、武器系统、核设施、生命支持设备或系统、其他医疗设备或系统（包括急救设备和手术植入物）、污染控制或有害物质管理系统中的关键部件，或产品植入之设备或系统故障可能导致人身伤害、死亡或财产损失其他用途（“非预期用途”）。关键部件指，若该部件发生故障，经合理预期会导致设备或系统故障或会影响设备或系统安全性和有效性的部件。针对由赛普拉斯产品非预期用途产生或相关的任何主张、费用、损失和其他责任，赛普拉斯不承担全部或部分责任且贵方不应追究赛普拉斯之责任。贵方应赔偿赛普拉斯因赛普拉斯产品任何非预期用途产生或相关的所有索赔、费用、损失和其他责任，包括因人身伤害或死亡引起的主张，并使之免受损失。

赛普拉斯、赛普拉斯徽标、Spansion、Spansion 徽标，及上述项目的组合，及 PSoC、CapSense、EZ-USB、F-RAM 和 Traveo 应视为赛普拉斯在美国和其他国家的商标或注册商标。请访问 cypress.com 获取赛普拉斯商标的完整列表。其他名称和品牌可能由其各自所有者主张为该方财产。

前言

感谢您继续使用 Cypress 产品。
使用此系列产品前请认真阅读本《手册》及《数据手册》。

本《手册》目的及目标读者

本《手册》说明本系列功能及操作，并说明如何使用。本《手册》专门设计给使用本系列产品实际开发的工程师使用。
有关模拟宏、计时器及通讯宏，参见相关独立的《外设手册》。

注意事项：

- *本《手册》说明外设功能的配置及操作，但不包括系列中各装置的具体内容。*
  有关各装置详细资料，用户可参阅相关《数据手册》。
- *板子是否支持外围功能基于芯片的类型，具体的细节需要参考使用芯片的数据手册。*

Trademark

ARM and Cortex are the trademarks of ARM Limited in the EU and other countries.

The company names and brand names herein are the trademarks or registered trademarks of their respective owners.

样本程序及开发环境

Cypress 免费提供使用 FM0+ 家族外设功能的样本程序。Cypress 还提供该系列所需开发环境的说明。可免费使用样本程序验证 Cypress 半导体微处理器的操作规范及使用情况。

微处理器支持信息：
http://www.spansion.com/support/microcontrollers/

注意事项：

- *请注意样本程序如有变更，恕不另行通知。样本程序是用于验证标准操作和使用情况，在使用前，请进行实际评估。*
  由于使用样本程序发生损坏，Cypress 不承担任何责任。

本《手册》的组织结构

《外设手册》计时器部分包括 14 章以及附录如下：

第 1 章：	系统概述
第 2-1 章：	时钟
第 2 -2 章：	外设时钟门控
第 2-3 章：	高速 CR 调节
第 2-4 章：	低速 CR 预分频器
第 3 章：	时钟监视器
第 4 章：	复位
第 5-1 章：	低压检测概述
第 5-2 章：	低压检测(TYPE1)
第 5-3 章：	低压检测(TYPE2)
第 5-4 章：	低压检测(TYPE 3)
第 6-1 章：	低功耗模式

第 6-2 章： Vbat 域
第 7-1 章： 中断概述
第 7-2 章： 中断(TYPE 1)
第 7-3 章： 中断(TYPE 1-A)
第 7-4 章： 中断(TYPE 1-B)
第 7-5 章： 中断(TYPE 2)
第 7-6 章： 中断(TYPE 2-A)
第 7-7 章： 中断(TYPE 2-B)
第 7-8 章： 中断(TYPE 3)
第 8 章： 外部中断及 NMI 控制段
第 9 章： DMAC
第 10-1 章： I/O 端口
第 10-2 章： 高速 GPIO
第 11 章： CRC （循环冗余检验）
第 12 章： 调试接口
第 13 章： 微跟踪缓冲数据监视点及跟踪
第 14 章： 闪存
第 15 章： 唯一 ID 寄存器
第 16 章： DSTC
附录

相关手册

此产品家族相关手册如下。参见相应条件相关手册。

这些手册中的内容如有变更，恕不另行通知。有关最新版本情况，请与我们联系。

外设手册

- FM0+家族外设手册（本手册）
  （以下称 “外设手册”）
- FM0+家族外设手册计时器部分
  （以下称“计时器部分”）
- FM0+家族外设手册模拟宏部分
  （以下称“模拟宏部分”）
- FM0+家族外设手册通讯宏部分
  （以下称“通讯宏部分”）

数据手册

有关具体设备的电气特性、封装尺寸以及订购信息等，参见以下文件。

- 32-bit 微处理器 FM0+ 家族数据手册

注意事项：

- *提供各系列数据手册。*
  参见您所使用系列的相关《数据手册》。

CPU 编程手册

有关 RM Cortex-M0+ 内核，参见可通过 http://www.arm.com/的以下文件：

- Cortex-M0+ 技术参考手册
- ARMv6-M 体系结构应用级参考手册

闪存编程手册

有关内置闪存的功能及操作，参见以下文件：

- FM0+ 家族闪存编程手册

注意事项：

- *提供各系列闪存编程手册。*
  参见您所使用产品系列的相关《闪存编程手册》。

如何使用本《手册》

功能查找

可采用以下方法搜寻本《手册》中相关功能的说明：

■ 搜寻目录

目录中按顺序列出《手册》内容。

■ 搜寻寄存器

文本中未描述各寄存器所在地址。核实寄存器地址，参见“附录”中“A.寄存器映射表"。

章节说明

本《手册》基本上按每章说明 1 个外设功能。

术语

本《手册》采用以下术语

术语	解释
字	指存取单位为 32 位。
半字	指存取单位为 16 位。
字节	指存取单位为 8 位。

符号说明:

■ 本《手册》寄存器解释的位配置符号写法如下：

- bit : 位编号
- Field : 位字段名
- Attribute : 各位的读写属性
- R : 只读
- W : 只写
- R/W : 可读/可写
- - : 未定义
- Initial value : 寄存器复位后的初始值
- 0 : 初始值为 "0"
- 1 : 初始值为 "1"
- X : 初始值未定义

■ 本《手册》中多位写法如下：

例如：bit7:0 表示 bit7 至 bit0 的数位。

■ 本《手册》中地址等的值按如下方式写入：

- 十六进制数 : 值前加前缀 "0x" （如：0xFFFF）
- 二进制数 : 值前加前缀 "0b" （如：0b1111）
- 十进制数 : 只用数字写（如：1000）

本《手册》目标产品

■ 本《手册》中，产品分类及描述如下。

有关 "TYPE 1-M0+"、 “TYPE 2-M0+” 和 “TYPE 3-M0+” 的描述，参见下表中目标 FM0+ 产品相关项如下：

Table 1FM0+ 家族 TYPE 1 产品表

类型	闪存容量	
	88 K 字节	**56 K 字节**
TYPE 1-M0+	S6E1A12B0A S6E1A12C0A	S6E1A11B0A S6E1A11C0A

Table 2FM0+ 家族 TYPE 2 产品表

类型	闪存容量		
	304K 字节	432K 字节	560K 字节
TYPE 2-M0+	S6E1B84xH (x=E,F,G) S6E1B74xH (x=E,F,G) S6E1B34xH (x=E,F,G) S6E1B14xH (x=E,F,G) S6E1B84x0 (x=E,F,G) S6E1B74x0 (x=E,F,G) S6E1B34x0 (x=E,F,G) S6E1B14x0 (x=E,F,G)	S6E1B85xH (x=E,F,G) S6E1B75xH (x=E,F,G) S6E1B35xH (x=E,F,G) S6E1B15xH (x=E,F,G) S6E1B85x0 (x=E,F,G) S6E1B75x0 (x=E,F,G) S6E1B35x0 (x=E,F,G) S6E1B15x0 (x=E,F,G)	S6E1B86xH (x=E,F,G) S6E1B76xH (x=E,F,G) S6E1B36xH (x=E,F,G) S6E1B16xH (x=E,F,G) S6E1B86x0 (x=E,F,G) S6E1B76x0 (x=E,F,G) S6E1B36x0 (x=E,F,G) S6E1B16x0 (x=E,F,G)

Table 3FM0+ 家族 TYPE 3 产品表

类型	闪存容量	
	64K 字节	128K 字节
TYPE 3-M0+	S6E1C31B0A S6E1C31C0A S6E1C31D0A S6E1C11B0A S6E1C11C0A S6E1C11D0A	S6E1C32B0A S6E1C32C0A S6E1C32D0A S6E1C12B0A S6E1C12C0A S6E1C12D0A

目录

第 1 章：系统概述 23
1. 总线结构 24
1.1 总线框图 25
1.2 存储器体系结构 26
1.3 存储器映射图 27
1.4 外设地址映射 29
2. Cortex-M0+ 结构 31
2.1 选项配置 33
3. 模式 34
第 2-1 章：时钟 35
1. 概述 36
2. 配置 37
3. 操作 40
3.1 选择时钟模式 40
3.2 内部总线时钟分频控制 41
3.3 PLL 时钟控制 42
3.4 振荡稳定等待时间 44
3.5 中断因素 47
4. 时钟设置步骤示例 48
4.1 设置步骤示例（TYPE 1-M0+、TYPE 3-M0+） 48
4.2 设置步骤示例（TYPE 2-M0+） 51
5. 寄存器 56
5.1 系统时钟模式控制寄存器 (SCM_CTL) 57
5.2 系统时钟模式状态寄存器 (SCM_STR) 59
5.3 基本时钟预分频寄存器 (BSC_PSR) 61
5.4 APB0 预分频寄存器 (APBC0_PSR) 62
5.5 APB1 预分频寄存器 (APBC1_PSR) 63
5.6 软件看门狗时钟预分频寄存器 (SWC_PSR) 64
5.7 时钟稳定等待时间寄存器(CSW_TMR) 65
5.8 PLL 时钟稳定等待时间设置寄存器 (PSW_TMR) 67
5.9 PLL 控制寄存器 1 (PLL_CTL1) 68
5.10 PLL 控制寄存器 2 (PLL_CTL2) 69
5.11 调试断点看门狗计时器控制寄存器 (DBWDT_CTL) 70

5.12 中断使能寄存器(INT_ENR) 71
5.13 中断状态寄存器(INT_STR) 72
5.14 中断清除寄存器(INT_CLR) 73
6. 使用注意事项 75
第 2-2 章：外设时钟门控 77
1. 外设时钟门控概述 78
2. 外设时钟门控配置 81
3. 外设时钟门控 84
3.1 外设时钟控制步骤 85
4. 外设时钟门控功能寄存器 89
4.1 外设功能时钟控制寄存器 0 (CKEN0) 90
4.2 外设复位控制寄存器 0 (MRST0) 93
4.3 外设时钟控制寄存器 1 (CKEN1) 97
4.4 外设功能复位控制寄存器 1 (MRST1) 99
4.5 外设时钟控制寄存器 2 (CKEN2) 101
4.6 外设功能复位控制寄存器 2 (MRST2) 105
5. 外设时钟门控功能使用注意事项 109
第 2-3 章：高速 CR 调节 113
1. 高速 CR 调节功能概述 114
2. 高速 CR 调节功能配置及框图 115
3. 高速 CR 调节功能操作 116
4. 高速 CR 调节功能设置步骤示例 117
5. 高速 CR 调节功能寄存器表 127
5.1 高速 CR 振荡分频设置寄存器 (MCR_PSR) 128
5.2 高速 CR 振荡频率调节寄存器 (MCR_FTRM) 129
5.3 高速 CR 振荡温度调节设置寄存器 (MCR_TTRM) 130
5.3.1 MCR_TTRM （TYPE 1-M0+） 130
5.3.2 MCR_TTRM（TYPE 2-M0+） 131
5.3.3 MCR_TTRM（TYPE 3-M0+） 132
5.4 高速 CR 振荡寄存器写保护寄存器 (MCR_RLR) 133
6. 高速 CR 调节功能使用注意事项 134
第 2-4 章:低速 CR 预分频器 135
1. 低速 CR 预分频器概述 136
2. 低速 CR 预分频器配置 137
3. 低速 CR 预分频器操作及设置步骤示例 138
3.1 低速 CR 预分频器设置步骤 138
3.2 低速 CR 预分频器操作 139
3.3 低速 CR 校正示例 140
4. 低速 CR 预分频寄存器 141
4.1 低速 CR 预分频器控制寄存器 (LCR_PRSLD) 142
第 3 章:时钟监视器 143
1. 概述 144
2. 配置及框图 145
2.1 时钟故障检测 145
2.2 异常频率检测 146
3. 操作说明 147
4. 设置步骤示例 148
5. 操作示例 150

5.1 时钟故障检测 150
5.2 异常频率检测 152
5.3 异常频率检测功能窗口设置示例 153
6. 寄存器 155
6.1 CSV 控制寄存器 (CSV_CTL) 156
6.2 CSV 状态寄存器 (CSV_STR) 158
6.3 频率检测窗口设置寄存器（上限）(FCSWH_CTL) 159
6.4 频率检测窗口设置寄存器（下限）(FCSWL_CTL) 160
6.5 频率检测计数器寄存器 (FCSWD_CTL) 161
7. 使用注意事项 162
第 4 章:复位 163
1. 概述 164
2. 配置 165
3. 操作说明 167
3.1 复位因素 168
3.2 设备内复位 171
3.2.1 复位至 Cortex-M0+ 172
3.2.2 复位至外设电路 173
3.3 复位顺序 174
3.4 清除复位后的操作 176
4. 寄存器 177
4.1 复位因素寄存器 (RST_STR) 178
第 5-1 章：低压检测概述 181
1. 概述 182
第 5-2 章：低压检测(TYPE 1) 185
1. 配置 186
2. 操作 188
3. 设置步骤示例 191
4. 寄存器 193
4.1 低压检测电压控制寄存器 (LVD_CTL) 194
4.2 低压检测中断因素寄存器 (LVD_STR) 196
4.3 低压检测中断因素清除寄存器 (LVD_CLR) 197
4.4 低压检测电压保护寄存器 (LVD_RLR) 198
4.5 低压检测电路状态寄存器(LVD_STR2) 199
5. 使用注意事项 200
第 5-3 章：低压检测(TYPE 2) 201
1. 配置 202
2. 操作 206
3. 设置步骤示例 209
3.1 低压检测复位示例 209
3.2 低压检测中断示例 (LVDRLIE = 0) 210
3.3 低压检测中断示例 (LVDRLIE = 1) 211
3.4 Vref 校准示例 213
4. 寄存器 214
4.1 低压检测电压控制寄存器 (LVD_CTL) 215
4.2 低压检测释放电压控制寄存器 (LVD_CTL2) 217
4.3 低压检测(ch.2)电压控制寄存器 (LVD2_CTL) 219
4.4 低压检测(ch.2)释放电压控制寄存器(LVD2_CTL2) 221

4.5 低压检测中断因素寄存器 (LVD_STR) 223
4.6 低压检测中断因素清除寄存器 (LVD_CLR) 225
4.7 低压检测电压保护寄存器 (LVD_RLR) 226
4.8 低压检测电路状态寄存器(LVD_STR2) 227
4.9 Vref 校准控制寄存器 (CAL_CTL) 229
4.10 Vref 校准值寄存器(CAL_SET) 230
4.11 Vref 校准密钥寄存器(CAL_KEY) 231
5. 使用注意事项 232
第 5-4 章：低压检测(TYPE 3) 233
1. 配置 234
2. 操作 236
3. 设置步骤示例 239
3.1 低压检测复位示例 239
3.2 低压检测中断示例 240
3.3 Vref 校准示例 241
4. 寄存器 242
4.1 低压检测电压控制寄存器 (LVD_CTL) 243
4.2 低压检测中断因素寄存器 (LVD_STR) 245
4.3 低压检测中断因素清除寄存器 (LVD_CLR) 246
4.4 低压检测电压保护寄存器 (LVD_RLR) 247
4.5 低压检测电路状态寄存器(LVD_STR2) 248
4.6 Vref 校准控制寄存器 (CAL_CTL) 249
4.7 Vref 校准密钥寄存器(CAL_KEY) 250
5. 使用注意事项 251
第 6-1 章：低功耗模式 253
1. 概述 254
2. CPU 操作模式配置 258
3. 待机模式的操作 270
3.1 休眠模式(高速 CR 休眠、主休眠、PLL 休眠、低速 CR 休眠和副休眠模式)操作 273
3.2 计时器模式(高速 CR 计时器、主计时器、PLL 计时器、低速 CR 计时器和副计时器模式)操作 275
3.3 RTC 模式操作 277
3.4 停止模式操作 280
4. 待机模式设置步骤示例 283
5. 深度待机模式操作说明 286
5.1 深度待机模式操作 286
5.1.1 深度待机模式的时钟操作状态 286
5.1.2 深度待机模式的唤醒因素 286
5.1.3 深度待机模式的内部电源状态及复位状态 287
5.2 深度待机 RTC 模式的操作 287
5.3 深度待机停止模式操作 290
6. 深度待机模式设置步骤示例 292
7. 深度待机唤醒因素确定步骤 294
8. 寄存器 295
8.1 待机模式控制寄存器 (STB_CTL) 296
8.2 副振荡电路电源控制寄存器 (REG_CTL) 298
8.3 副时钟控制寄存器 (RCK_CTL) 299
8.4 RTC 模式控制寄存器(PMD_CTL) 300

8.5 深度待机唤醒因素寄存器 1 (WRFSR) 301
8.6 深度待机唤醒因素寄存器 2 (WIFSR) 302
8.7 深度待机唤醒使能寄存器 (WIER) 304
8.8 WKUP 引脚输入电平寄存器 (WILVR) 306
8.9 深度待机 RAM 保留寄存器 (DSRAMR) 307
8.10 备用寄存器 01 至 16 (BUR01 至 16) 308
8.11 主晶振型选择控制寄存器 (MOSC_CTL) 309
8.12 IO 状态保持控制寄存器 (WIOLC_CTL) 310
8.13 副振荡器 IO 控制寄存器 (SUBOSC_CTL) 312
8.14 CEC 输入/输出控制寄存器 (CEC_CTL) 313
8.15 串行线调试控制寄存器 (DEBUG_SW_CTL) 314
9. 使用注意事项 315
第 6-2 章：Vbat 域 317
1. VBAT 域概述 318
2. VBAT 域配置 320
2.1 与常开域的接口 321
2.2 RTC 331
2.3 32 kHz 振荡电路 332
2.4 上电电路 334
2.5 备用寄存器 335
2.6 VBAT I/O 端口 337
3. 芯片电源控制 340
4. 睡眠控制 344
5. 32 kHz 时钟设置步骤 348
6. VBAT I/O 端口设置步骤 352
7. 寄存器 355
7.1 VB_CLKDIV 寄存器 356
7.2 WTOSCCNT 寄存器 357
7.3 CCS/CCB 寄存器 358
7.4 BOOST 寄存器 359
7.5 EWKUP 寄存器 360
7.6 HIBRST 寄存器 361
7.7 VDET 寄存器 362
7.8 端口功能设置寄存器 (VBPFR) 363
7.9 上拉设置寄存器 (VBPCR) 365
7.10 端口 I/O 方向设置寄存器 (VBDDR) 366
7.11 端口输入数据寄存器 (VBDIR) 367
7.12 端口输出数据寄存器 (VBDOR) 368
7.13 端口伪开漏设置寄存器 (VBPZR) 369
8. 使用注意事项 370
第 7-1 章：中断概述 371
1. 中断参考章节说明 372
第 7-2 章：中断(TYPE 1) 373
1. 概述 374
2. 配置 375
第 7-3 章：中断(TYPE 1-A) 379
1. 异常及中断因素向量 380
2. 寄存器 382

2.1 DMA 请求选择寄存器(DRQSEL)..... 383
2.2 EXC02 批量读取寄存器 (EXC02MON)..... 385
2.3 IRQ00 批量读取寄存器(IRQ00MON)..... 386
2.4 IRQ01 批量读取寄存器(IRQ01MON)..... 387
2.5 IRQ02 批量读取寄存器 (IRQ02MON)..... 388
2.6 IRQ03 批量读取寄存器 (IRQ03MON)..... 389
2.7 IRQ04 批量读取寄存器 (IRQ04MON)..... 391
2.8 IRQ05 批量读取寄存器 (IRQ05MON)..... 392
2.9 IRQ06 批量读取寄存器 (IRQ06MON)..... 394
2.10 IRQ07/09/11/13/15/17 批量读取寄存器(IRQxxMON)..... 396
2.11 IRQ19/21 批量读取寄存器 (IRQxxMON)..... 397
2.12 IRQ08/10/12/14/16/18 批量读取寄存器 (IRQxxMON)..... 398
2.13 IRQ20/22 批量读取寄存器 (IRQxxMON)..... 399
2.14 IRQ23 批量读取寄存器 (IRQ23MON)..... 400
2.15 IRQ24 批量读取寄存器 (IRQ24MON)..... 401
2.16 IRQ25/26 批量读取寄存器(IRQxxMON)..... 403
2.17 IRQ27 批量读取寄存器(IRQ27MON)..... 404
2.18 IRQ28 批量读取寄存器 (IRQ28MON)..... 405
2.19 IRQ29 批量读取寄存器 (IRQ29MON)..... 407
2.20 IRQ30 批量读取寄存器 (IRQ30MON)..... 409
2.21 IRQ31 批量读取寄存器 (IRQ31MON)..... 411
3. 使用注意事项..... 413
第 7-4 章：中断(TYPE 1-B)..... 415
1. 异常及中断因素向量..... 416
2. 寄存器..... 418
2.1 DMA 请求选择寄存器(DRQSEL)..... 420
2.2 EXC02 批量读取寄存器 (EXC02MON)..... 422
2.3 IRQ00 批量读取寄存器(IRQ00MON)..... 423
2.4 IRQ01 批量读取寄存器(IRQ01MON)..... 424
2.5 IRQ02 批量读取寄存器 (IRQ02MON)..... 425
2.6 IRQ03~IRQ10 批量读取寄存器 (IRQ03MON ~ IRQ10MON)..... 426
2.7 IRQ11 批量读取寄存器 (IRQxxMON)..... 427
2.8 IRQ12 批量读取寄存器 (IRQ12MON)..... 428
2.9 IRQ13 批量读取寄存器 (IRQ13MON)..... 429
2.10 IRQ14 批量读取寄存器 (IRQ14MON)..... 431
2.11 IRQ15 ~ IRQ18 批量读取寄存器 (IRQ15MON ~ IRQ18MON)..... 432
2.12 IRQ19/21 批量读取寄存器 (IRQxxMON)..... 433
2.13 IRQ20/22 批量读取寄存器 (IRQxxMON)..... 434
2.14 IRQ23 批量读取寄存器 (IRQ23MON)..... 435
2.15 IRQ24 批量读取寄存器 (IRQ24MON)..... 436
2.16 IRQ25/26 批量读取寄存器(IRQxxMON)..... 438
2.17 IRQ27 批量读取寄存器(IRQ27MON)..... 440
2.18 IRQ28/29 批量读取寄存器 (IRQxxMON)..... 442
2.19 IRQ30 批量读取寄存器 (IRQ30MON)..... 444
2.20 IRQ31 批量读取寄存器 (IRQ31MON)..... 446
2.21 中断因素向量重定位设置寄存器 (IRQCMODE)..... 448
2.22 中断因素选择寄存器 0 (RCINTSEL0)..... 449
2.23 中断因素选择寄存器 1 (RCINTSEL1)..... 450

3. 使用注意事项 453
第 7-5 章：中断(TYPE 2) 455
1. 概述 456
2. 配置 457
第 7-6 章：中断(TYPE 2-A) 461
1. 中断列表 462
1.1 异常及中断列表 462
1.2 输入至 DSTC 的中断信号 467
2. 寄存器 469
2.1 EXC02 批量读取寄存器 (EXC02MON) 470
2.2 IRQxx 批量读取寄存器 (IRQxxMON) 471
3. 使用注意事项 472
第 7-7 章：中断(TYPE 2-B) 473
1. 中断列表 474
1.1 异常及中断列表 474
1.2 输入至 DSTC 的中断信号 479
2. 寄存器 481
2.1 EXC02 批量读取寄存器 (EXC02MON) 482
2.2 IRQxx 批量读取寄存器 (IRQxxMON) 483
2.3 中断因素向量重定位设置寄存器 (IRQCMODE) 484
2.4 中断因素选择寄存器 0 (RCINTSEL0) 485
2.5 中断因素选择寄存器 1 (RCINTSEL1) 486
3. 使用注意事项 488
第 7-8 章：中断(TYPE 3) 489
1. 概述 490
2. 配置与功能 491
2.1 连接图 491
2.2 NVIC (嵌套向量中断控制器) 492
2.3 中断集合及批量读取寄存器 492
2.4 向量指示寄存器 492
2.5 DSTC 传输请求与传输完成的连接 494
2.6 DMA 传输确认信号的连接 494
2.7 DMA 传输停止信号的连接 494
3. 中断列表 495
3.1 异常及中断列表 495
3.2 输入至 DSTC 的传输请求信号列表 499
4. 寄存器 500
4.1 寄存器列表 500
4.2 EXC02 批量读取寄存器 (EXC02MON) 501
4.3 IRQxx 批量读取寄存器 (IRQxxMON) 502
4.4 VIR 偏移寄存器 (VIR_OFFSET) 503
4.5 向量指示寄存器 xx (VIRxx) 504
4.6 USB ch.0 奇数包长 DMA 使能寄存器 (ODDPKS) 505
5. 使用注意事项 507
第 8 章：外部中断及 NMI 控制段 509
1. 概述 510
2. 框图 511
3. 操作及设置步骤示例 512

3.1 外部中断控制段操作 513
3.2 NMI 控制段操作 515
3.3 从计时器、停止和 RTC 模式唤醒 516
4. 寄存器 517
4.1 外部中断使能寄存器 (ENIR) 518
4.2 外部中断因素寄存器 (EIRR) 519
4.3 外部中断因素清除寄存器 (EICL) 520
4.4 外部中断因素电平寄存器 (ELVR) 521
4.5 外部中断因素电平寄存器 1 (ELVR1) 522
4.6 非屏蔽中断因素寄存器 (NMIRR) 523
4.7 非屏蔽中断因素清除寄存器(NMICL) 524
4.8 外部中断因素电平寄存器 2 (ELVR2) 525
4.9 非屏蔽中断使能寄存器 (NMIENR) 526
第 9 章：DMAC 527
1. DMAC 概述 528
2. DMAC 配置 529
2.1 DMAC 和系统配置 530
2.2 DMAC 的 I/O 信号 532
3. DMAC 功能及操作 534
3.1 软件-块传输 535
3.2 软件-突发传输 537
3.3 硬件-命令传输 538
3.4 硬件-块传输及突发传输 539
3.5 通道优先控制 541
4. DMAC 控制 542
4.1 DMAC 控制概述 543
4.2 软件传输的 DMAC 操作及控制步骤 544
4.3 硬件 (EM=0) 传输的 DMAC 操作及控制步骤 553
4.4 硬件 (EM=1) 传输的 DMAC 操作及控制步骤 563
5. DMAC 寄存器 568
5.1 寄存器列表 569
5.2 完整 DMAC 配置寄存器 (DMACR) 570
5.3 配置 A 寄存器 (DMACA) 572
5.4 配置 B 寄存器 (DMACB) 575
5.5 传输源地址寄存器 (DMACSA) 579
5.6 传输目标地址寄存器 (DMACDA) 580
6. 使用注意事项 581
第 10-1 章:I/O 端口 583
1. 概述 584
2. 配置、框图及操作 585
2.1 I/O 端口配置 585
2.2 I/O 端口初选功能 590
2.3 重定位功能 591
2.4 EPFR 输出的固定优先级 593
2.5 深度待机模式（TYPE 2-M0+）下的操作 594
2.6 深度待机模式（TYPE 3-M0+产品）时的操作 595
2.7 深度待机转换复位 598
3. 设置步骤示例 599

3.1 I/O 端口设置 599
3.2 如何使用 I/O 状态保持功能（TYPE 3-M0+产品） 600
4. 寄存器 602
4.1 端口功能设置寄存器 (PFRx) 606
4.2 上拉设置寄存器 (PCRx) 608
4.3 端口输入/输出方向设置寄存器 (DDRx) 610
4.4 端口输入数据寄存器 (PDIRx) 612
4.5 端口输出数据寄存器 x (PDORx) 614
4.6 模拟输入设置寄存器 (ADE) 616
4.7 扩展引脚功能设置寄存器 (EPFRx) 617
4.8 扩展引脚功能设置寄存器 00(EPFR00) 619
4.9 扩展引脚功能设置寄存器 01 (EPFR01) 622
4.10 扩展引脚功能设置寄存器 02 (EPFR02) 627
4.11 扩展引脚功能设置寄存器 03 (EPFR03) 631
4.12 扩展引脚功能设置寄存器 04 (EPFR04) 635
4.13 扩展引脚功能设置寄存器 05 (EPFR05) 639
4.14 扩展引脚功能设置寄存器 06 (EPFR06) 643
4.15 扩展引脚功能设置寄存器 07 (EPFR07) 647
4.16 扩展引脚功能设置寄存器 08 (EPFR08) 652
4.17 扩展引脚功能设置寄存器 09 (EPFR09) 657
4.18 扩展引脚功能设置寄存器 12 (EPFR12) 662
4.19 扩展引脚功能设置寄存器 13 (EPFR13) 666
4.20 扩展引脚功能设置寄存器 14 (EPFR14) 670
4.21 扩展引脚功能设置寄存器 15 (EPFR15) 672
4.22 扩展引脚功能设置寄存器 16 (EPFR16) 676
4.23 扩展引脚功能设置寄存器 17 (EPFR17) 681
4.24 扩展引脚功能设置寄存器 18 (EPFR18) 686
4.25 扩展引脚功能设置寄存器 21 (EPFR21) 688
4.26 扩展引脚功能设置寄存器 22 (EPFR22) 690
4.27 扩展引脚功能设置寄存器 23 (EPFR23) 692
4.28 扩展引脚功能设置寄存器 31 (EPFR31) 695
4.29 扩展引脚功能设置寄存器 33 (EPFR33) 697
4.30 扩展引脚功能设置寄存器 34 (EPFR34) 701
4.31 扩展引脚功能设置寄存器 37 (EPFR37) 703
4.32 扩展引脚功能设置寄存器 38 (EPFR38) 707
4.33 专用端口设置寄存器 (SPSR) 709
4.34 端口伪开漏设置寄存器 (PZRx) 711
4.35 LVDI 输入设置寄存器 (LVDIE) 713
5. 使用注意事项 714
第 10-2 章：高速 GPIO 717
1. 概述 718
2. 配置 719
3. 设置步骤示例 725
4. 寄存器 727
4.1 高速 GPIO 输入数据寄存器(FPDIRx) 730
4.2 高速 GPIO 输出数据寄存器 x (FPDORx) 732
4.3 镜像高速 GPIO 输入数据寄存器(M_FPDIRx) 734
4.4 镜像高速 GPIO 输出数据寄存器(M_FPDORx) 736

4.5 高速 GPIO 输出使能寄存器(FPOERx) 738
5. 位处理基址 740
第 11 章：CRC（循环冗余校验） 741
1. CRC 概述 742
2. CRC 操作 743
2.1 CRC 计算序列 744
2.2 CRC 使用示例 745
3. CRC 寄存器 751
3.1 CRC 控制寄存器 (CRCCR) 752
3.2 初始值寄存器 (CRCINIT) 754
3.3 输入数据寄存器 (CRCIN) 755
3.4 CRC 寄存器 (CRCR) 756
第 12 章: 调试接口 757
1. 概述 758
2. 引脚描述 759
2.1 调试用引脚 760
2.2 引脚初始功能 761
2.3 SW-DP 引脚内部上拉 762
第 13 章: 微跟踪缓冲数据监视点及跟踪 763
1. 概述 764
2. 框图 765
3. 配置及设置步骤示例 766
3.1 MTB_DWT 配置 767
3.2 设置步骤 768
4. 寄存器 770
4.1 MTB_DWT 地址比较启动跟踪寄存器 (CMP_ADDR_START) 771
4.2 MTB_DWT 数据比较启动跟踪寄存器 (CMP_DATA_START) 772
4.3 MTB_DWT 屏蔽数据比较启动跟踪寄存器 (CMP_MASK_START) 773
4.4 MTB_DWT 地址比较停止跟踪寄存器 (CMP_ADDR_STOP) 774
4.5 MTB_DWT 数据比较停止跟踪寄存器 (CMP_DATA_START) 775
4.6 MTB_DWT 屏蔽数据比较停止跟踪寄存器 (CMP_MASK_STOP) 776
4.7 MTB_DWT 功能寄存器 (FCT) 777
4.8 外设 ID0-7 寄存器 (PID0-7) 779
4.9 元件 ID0-3 寄存器 (CID0-3) 780
第 14 章：闪存 781
第 15 章: 唯一 ID 寄存器 783
1. 概述 784
2. 寄存器 785
2.1 唯一 ID 寄存器 0(UIDR0: 唯一 ID 寄存器 0) 786
2.2 唯一 ID 寄存器 1(UIDR1: 唯一 ID 寄存器 1) 787
第 16 章： DSTC 789
1. DSTC 概述 790
2. DSTC 操作概述及 DSTC 系统配置 791
2.1 DSTC 操作概述 791
2.2 DSTC 和系统配置 793
3. DSTC 功能及操作 796
3.1 DES 设置 797
3.1.1 传输数据量的指定 797

3.1.2 传输地址设置 798
3.1.3 指定外部重载 800
3.1.4 设置链路启动及传输结束通知 803
3.1.5 其他 DES 设置 805
3.2 DSTC 的控制功能 808
3.2.1 DSTC 内部结构框图 808
3.2.2 DESTP 寄存器 808
3.2.3 SW 传输控制 808
3.2.4 HW 传输控制 809
3.2.5 传输请求的判优 809
3.2.6 读取跳转缓冲功能 811
3.2.7 传输结束操作 811
3.2.8 MONERS 寄存器 812
3.2.9 待机功能 814
3.3 DSTC 操作流程 816
3.3.1 SW 传输流程 816
3.3.2 HW 传输流程 820
3.3.3 指定 DESP 后的操作流程 823
4. DSTC 操作和控制示例 827
4.1 传输操作示例 1 828
4.2 传输操作示例 2 832
4.3 传输操作示例 3 835
4.4 传输操作示例 4 837
4.5 传输操作示例 5 840
4.6 DSTC 控制示例 844
5. DSTC 寄存器和描述符 847
5.1 控制寄存器及 DES 列表 848
5.2 DESTP 寄存器 849
5.3 HWDESP[n] 寄存器 850
5.4 CMD 寄存器 851
5.5 CFG 寄存器 852
5.6 SWTR 寄存器 854
5.7 MONERS 寄存器 855
5.8 DREQENB[n] 寄存器 858
5.9 HWINT[n] 寄存器 859
5.10 HWINTCLR[n] 寄存器 860
5.11 DQMSK[n]寄存器 861
5.12 DQMSKCLR[n] 寄存器 862
5.13 描述符 0 (DES0) 863
5.14 描述符 1 (DES1) 868
5.15 描述符 2 (DES2) 870
5.16 描述符 3 (DES3) 870
5.17 描述符 4 (DES4) 871
5.18 描述符 5 (DES5) 871
5.19 描述符 6 (DES6) 871
附录 873
A. 产品类型 875
1. 产品类型列表 876

B. 寄存器映射(TYPE1-M0+) 877
1. 寄存器映射 878
1.1 闪存 I/F 879
1.2 唯一 ID 879
1.3 时钟/复位 880
1.4 HW WDT 882
1.5 SW WDT 882
1.6 双计时器 883
1.7 MFT 884
1.8 PPG 888
1.9 基本计时器 892
1.10 基本计时器的 IO 选择器 893
1.11 QPRC 895
1.12 QPRC NF 896
1.13 A/DC 896
1.14 D/AC 897
1.15 CR 调节 898
1.16 EXTI 898
1.17 INT-Req. 读取 899
1.18 LCDC 901
1.19 GPIO 902
1.20 HDMI-CEC 911
1.21 LVD 912
1.22 DS 模式 912
1.23 MFS 913
1.24 CRC 914
1.25 计时计数器 914
1.26 RTC 915
1.27 低速 CR 预分频器 916
1.28 外设时钟选通 916
1.29 DMAC 916
1.30 MTB_DWT 919
1.31 快速 GPIO 921
C. 寄存器映射(TYPE 2-M0+) 925
1. 寄存器映射 926
1.1 闪存 I/F 927
1.2 唯一 ID 927
1.3 时钟/复位 928
1.4 HW WDT 930
1.5 SW_WDT 930
1.6 双计时器 932
1.7 MFT 933
1.8 PPG 937
1.9 基本计时器 943
1.10 基本计时器的 IO 选择器 944
1.11 QPRC 946
1.12 QPRC NF 947
1.13 A/DC 947

1.14 D/AC 948
1.15 CR Trim 949
1.16 EXTI 950
1.17 INT-Req. 读取 950
1.18 LCDC 953
1.19 GPIO 954
1.20 HDMI-CEC 964
1.21 LVD 965
1.22 DS 模式 966
1.23 USB 时钟 967
1.24 MFS 969
1.25 CRC 971
1.26 计时计数器 971
1.27 RTC 972
1.28 低速 CR 预分频器 976
1.29 外设时钟选通 976
1.30 智能卡 I/F 977
1.31 MFSI2S 978
1.32 高容错性 978
1.33 USB 979
1.34 DSTC 981
1.35 MTB_DWT 982
1.36 快速 GPIO 984
D. 寄存器映射(TYPE 3-M0+) 989
1. 寄存器映射 990
1.1 闪存 I/F 992
1.2 唯一 ID 992
1.3 时钟/复位 993
1.4 HW WDT 995
1.5 SW WDT 995
1.6 双计时器 996
1.7 基本计时器 997
1.8 基本计时器的 IO 选择器 998
1.9 A/DC 999
1.10 CR 调节 1000
1.11 EXTI 1000
1.12 INT-Req. 读取 1002
1.13 GPIO 1004
1.14 HDMI-CEC 1010
1.15 LVD 1011
1.16 DS 模式 1012
1.17 USB 时钟 1013
1.18 I2CSLAVE 1014
1.19 MFS 1014
1.20 CRC 1016
1.21 计时计数器 1016
1.22 RTC 1017
1.23 低速 CR 预分频器 1018

1.24 外设时钟选通 1018
1.25 智能卡 I/F 1019
1.26 MFSI2S 1020
1.27 USB 1021
1.28 DSTC 1023
1.29 MTB_DWT 1024
1.30 快速 GPIO 1026
1.31 VIR 1028
E. 注意事项列表 1031
1. 高速 CR 用为主控时钟时的注意事项 1032
Revision History 1033

第 1 章 系统概述

本章为系统概述。

1. 总线结构
2. Cortex-M0+ 结构
3. 模式

代码：9AFSYSTEM-C03.0

1. 总线结构

本节说明总线结构。

对于本串行总线，AHB 总线矩阵电路采用多层总线。主机及从机体系结构如下：

■ 主结构
- Cortex-M0+ CPU(AHB-Lite)
- DMAC
- DSTC

■ 从属结构
- 片上闪存
- 片上 SRAM（可复用 MTB）
- AHB-AHB 总线桥
- AHB-APB 总线桥 (APB0、 APB1)
- USB ch.0/ch.1

总线框图参见 Figure 1-1 。

特征

■ RAM 体系结构

用户 SRAM 区可与 MTB SRAM 区共用，这两个区根据用户配置划分。

■ APB 扩展总线

APB1 外设总线为 APB 扩展总线，该总线原来在 AMBA3.0 的基础上添加了以下功能（不包括 APB0）。

- 支持半字（16 位）访问和字节（8 位）访问

  对于支持的寄存器，允许进行半字访问和字节访问。
  有关支持寄存器的信息，参见“附录”部分的“A. 寄存器映射图”。

- 添加“读-改-写(RMW)”信号

  使用位带操作的 HMASTLOCK 信号生成。
  RMW 信号的作用是防止在位带操作的读-改-写进程中误清除不相关标志。
  读-改-写进程中对应标志读为 "1" ，设计上忽略写入 "1"。
  这样可以防止按读-改-写顺序读取后立即设置标志时被下一次写入操作误清除不相关标志。
  对于对应的标志和寄存器，描述为“不考虑位值”， 在“读-改-写”进程中读为 "1"。

注意事项：
- *禁止 RMW 的寄存器不能执行位带操作。*
- *在不带位带操作功能的软件上执行读-改-写进程时，RMW 信号不输出。*
  因此，在这种情况下，标志值可通过支持 RMW 进程的寄存器进行读取，但要注意不要在写入操作过程误清除掉不相关标志。
- *有关位带操作的详细信息，参见《Cortex-M3 技术参考手册》，因为 Cortex-M3 支持位带操作。*

■ 优先级

总线权限的优先级按照循环法确定。

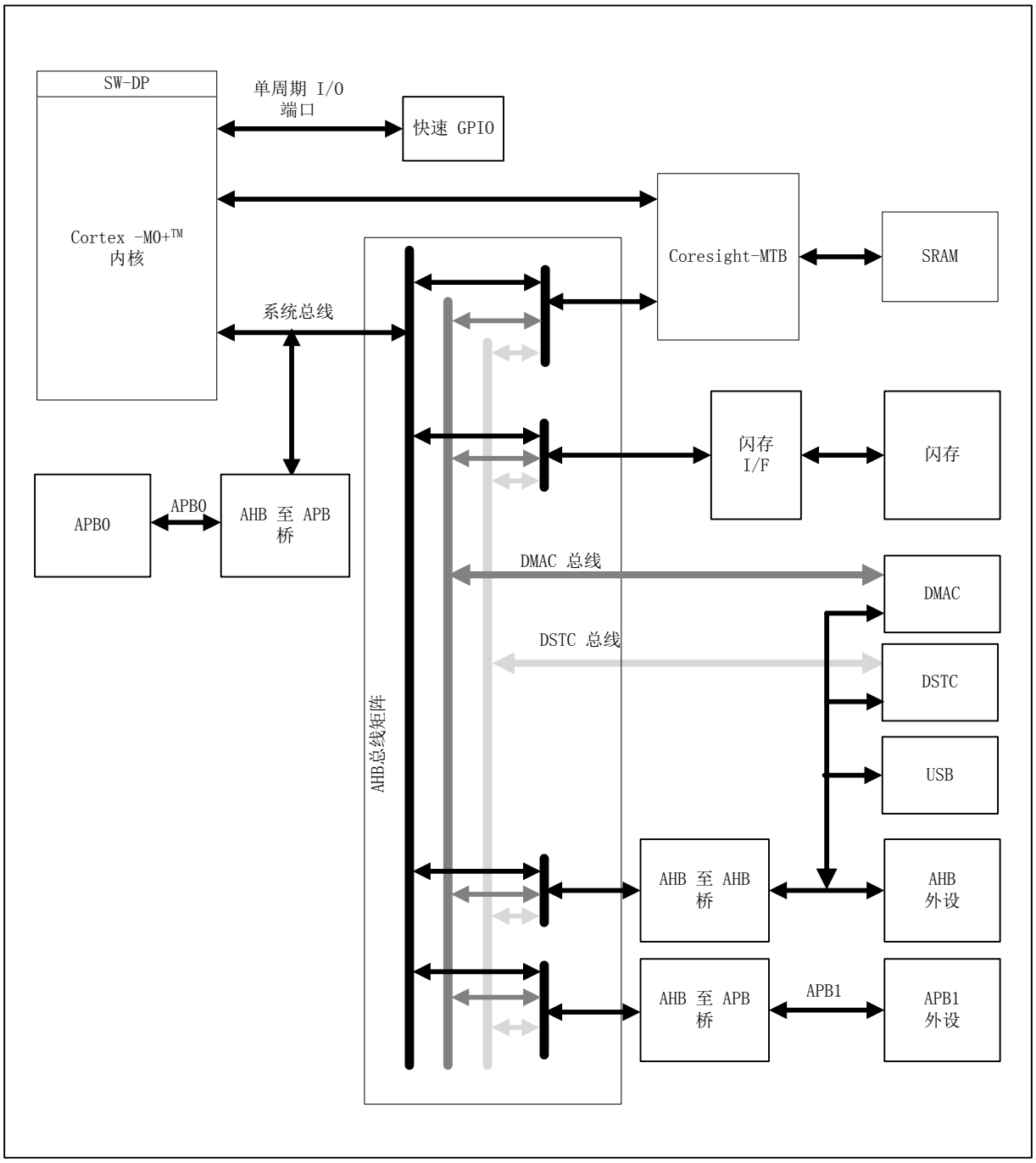
■ 字节端

本产品族采用小端字节序。

1.1 总线框图

Figure 1-1 所示为总线框图。

Figure 1-1 总线框图



注意事项：

- *有些区域无法执行 DMA 传输，详细信息参见 "1.3 存储器映射图" 以及 Table 1-1 的“DMA 传输”一栏。*

1.2 存储器体系结构

本节描述存储器的体系结构。

本产品族有 4 GB 地址空间。

闪存区和 SRAM 区定义最大容量分别为 1 MB 和 512KB。

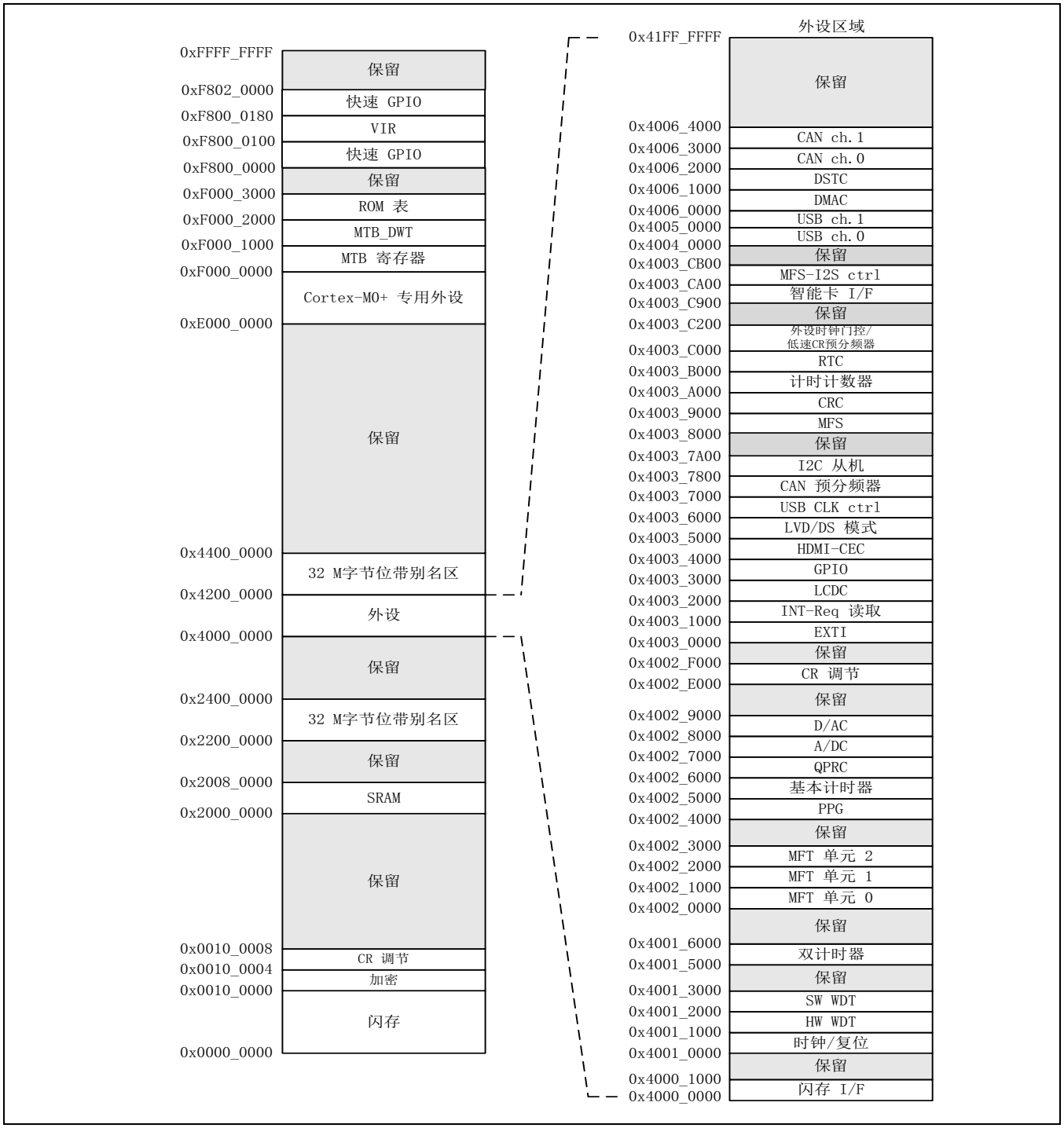
存储器映射参见 "1.3 存储器映射"一节，外设地址映射 "1.4 外设地址映射"一节。

如需了解 Figure 1-2 中所示的 Cortex-M0+专用外设的详细信息，参见《Cortex-M0+ 技术参考手册》。

1.3 存储器映射图

Figure 1-2 所示为存储器映射图。

Figure 1-2 存储器映射图



注意事项：

- *不可访问保留区。*
- *闪存详细信息，参见所用产品的《闪存编程手册》。*
- *不能对以下区域执行 DMA 传输：*
  - *位带别名区*
  - *快速 GPIO*
  - *VIR*
  - *ROM 表*
  - *MTB_DWT*
  - *MTB 寄存器 (SFR)*
  - *Coretex-M0+ 专用外设*

1.4 外设地址映射

Table 1-1 所列为外设地址映射。

Table 1-1 外设地址映射

起始地址	结束地址	总线	DMA 传输	外设	寄存器映射	章节
0x4000_0000	0x4000_0FFF	AHB	禁用	闪存 IF 寄存器/ 唯一 ID 寄存器	FLASH_IF / 唯一 ID	* 第 15 章
0x4000_1000	0x4000_FFFF			保留	-	-
0x4001_0000	0x4001_0FFF	APB0	禁用	时钟和复位控制	时钟 / 复位	第 2-1 章 第 3 章 第 4 章
0x4001_1000	0x4001_1FFF			硬件看门狗计时器	HWWDT	“计时器部分”的 第 1 章
0x4001_2000	0x4001_2FFF			软件看门狗计时器	SWWDT	
0x4001_3000	0x4001_4FFF			保留	-	-
0x4001_5000	0x4001_5FFF			双计时器	双计时器	“计时器部分”第 2 章
0x4001_6000	0x4001_FFFF			保留	-	-
0x4002_0000	0x4002_0FFF	APB1	使能	多功能计时器单元 0	MFT	“计时器部分”第 6 章
0x4002_1000	0x4002_1FFF			多功能计时器单元 1	MFT	
0x4002_2000	0x4002_2FFF			多功能计时器单元 2	MFT	
0x4002_3000	0x4002_3FFF			保留	-	-
0x4002_4000	0x4002_4FFF			PPG	PPG	“计时器部分”第 7-1、7-2 及 7-3 章
0x4002_5000	0x4002_5FFF			基本计时器	基本计时器/ 基本计时器 选择器	“计时器部分”第 5-1 和 5-2 章
0x4002_6000	0x4002_6FFF			QPRC	QPRC	“计时器部分”第 8-1 和 8-2 章
0x4002_7000	0x4002_7FFF			A/D 转换器	A/DC	“模拟宏部分”第 1 章
0x4002_8000	0x4002_8FFF			D/A 转换器	D/AC	“模拟宏部分”第 2 章
0x4002_9000	0x4002_DFFF			保留	-	-
0x4002_E000	0x4002_EFFF			高速 CR 调节	CR 调节	第 2-3 章
0x4002_F000	0x4002_FFFF			保留	-	-

起始地址	结束地址	总线	DMA 传输	外设	寄存器映射	章节
0x4003_0000	0x4003_0FFF	APB1	使能	外部中断	EXTI	第 8 章
0x4003_1000	0x4003_1FFF			中断源检查寄存器	INT-Req READ	第 7 章
0x4003_2000	0x4003_2FFF			LCDC	LCDC	“模拟宏部分”第 3 章
0x4003_3000	0x4003_3FFF			GPIO	GPIO	第 10 章
0x4003_4000	0x4003_4FFF			HDMI-CEC/远控接收器	HDMI-CEC	“通信宏部分”第 3 章
0x4003_5000	0x4003_50FF			低压检测	LVD	第 5 章
0x4003_5100	0x4003_5FFF			深度待机控制模块	DS_Mode	第 6 章
0x4003_6000	0x4003_6FFF			USB 时钟生成模块/USB	USB 时钟	“通信宏部分”第 4 章
0x4003_7000	0x4003_77FF			CAN 预分频器	CAN_ 预分频器	“通信宏部分”第 2-1 章
0x4003_7800	0x4003_79FF			I2C 从机唤醒	I2CSLAVE	“通信宏部分”第 7 章
0x4003_7A00	0x4003_7FFF			保留	-	-
0x4003_8000	0x4003_8FFF			多功能串口	MFS	“通信宏部分”第 1-2、1-3、1-4 及 1-5 章
0x4003_9000	0x4003_9FFF			CRC	CRC	第 11 章
0x4003_A000	0x4003_AFFF			计时计数器	计时计数器	“计时器部分”第 3 章
0x4003_B000	0x4003_BFFF			实时时钟	RTC	“计时器部分”第 4 章
0x4003_C000	0x4003_C1FF			外设时钟门控/低速 CR 预分频器	外设时钟门控	第 2-2 章 第 2-4 章
0x4003_C200	0x4003_C8FF			保留	-	-
0x4003_C900	0x4003_C9FF			IC 卡（智能卡）接口	智能卡 I/F	“通信宏部分”第 6 章
0x4003_CA00	0x4003_CAFF			I^2S 时钟生成器	MFS-I2S ctrl	“通信宏部分”第 1-6 章
0x4003_CB00	0x4003_FFFF			保留	-	-
0x4004_0000	0x4005_FFFF	AHB	使能	USB ch.0/ch.1	USB	“通信宏部分”第 5 章
0x4006_0000	0x4006_0FFF			DMAC	DMAC	第 9 章
0x4006_1000	0x4006_1FFF			DSTC	DSTC	第 16 章
0x4006_2000	0x4006_2FFF			CAN ch.0	CAN	“通信宏部分”第 2 章
0x4006_3000	0x4006_3FFF			CAN ch.1	CAN	
0x4006_4000	0x41FF_FFFF			保留	-	-

*: 有关“闪存 IF 寄存器”的详细信息，参见所用产品的《闪存编程手册》。

2. Cortex-M0+ 结构

本节描述本产品族所用的内核架构。

本产品族所用的 Cortex-M0+ 内核区架构如下：

- Cortex-M0+内核
- NVIC
- 数据观察点单元
- BPU
- MTB
- SW-DP
- ROM 表
- 单周期 I/O 端口

*：产品架构各不相同，详见“2.1 选项配置”。

Cortex-M0+内核

本产品族配有高效能 32 位处理器内核（ARM Cortex-M0+内核）。
本《外设手册》不对 Cortex-M0+内核做详细说明。
详细信息参见《Cortex-M0+技术参考手册》。

- Cortex-M0+ 内核版本

Cortex-M0+ 内核版本的详细信息参见所用产品的《数据手册》。

NVIC（嵌套向量中断控制器）

在本系列产品中，可使用一个 NMI（非屏蔽中断）和最多 32 个外设中断(IRQ0 至 IRQ31)*[1]。
此外，中断优先寄存器（地址：0xE000E400）为 2 位，可配置 4 个中断优先级。

如需了解外设中断的详细信息，参见所用产品手册的“中断”一章中的“中断配置”；有关 NMI 操作的详细信息，也请参考另外一章“外部中断和非屏蔽中断控制模块”。

NMIX 引脚复用通用端口。 重置后，引脚用作通用端口（初始功能），此时 NMI 中断被屏蔽。 使用 NMI 功能时，需通过端口设置使能 NMI 功能。 详细信息参见“I/O 端口”及“外部中断和 NMI 控制器”这些章节。

*1: 《Cortex-M0+技术参考手册》中将异常类型 IRQ 定义为外部中断。
为了与外部引脚“外部中断和 NMI 控制模块”引起的中断区分开，本《外设手册》中将异常类型 IRQ 定义为外设中断。

当其它优先级更高的中断请求正在处理时，其它外设功能向 NVIC 发出的中断请求会在 NVIC 中挂起。 取消 NVIC 中挂起的中断请求时，应从外设功能清除中断请求，然后通过 NVIC 中安装的中断清除挂起寄存器（地址：0xE000E280）清除 NVIC 中挂起的中断请求。

SysTick 计时器

SysTick 计时器是用于 NVIC 集成 OS 任务管理的系统计时器。

本产品族中，STCLK 是通过 HCLK 除以 8 生成，SysTick 校准值寄存器（地址：0xE000E01C）的值设置如下：

bit31 : NOREF = 0
bit30 : SKEW = 1
bit23:0 : TENMS =0x00C350(50000)*1

*1： TENMS 值的设置应当是当 HCLK 的 1/8 时钟输入至 STCLK 且 HCLK 频率为 40 MHz 时（ 1/8 时的 5 MHz），其值为 10 ms。
TENMS 的值并不总为 10ms，因为 HCLK 的频率可通过时钟控制模块更改。 因此需要根据 HCLK 的频率计算出适当的中断时间参数。

数据观察点单元

本产品族配有观察点单元，用于除错功能。

观察点单元由两个比较器组成，可用作数据地址匹配和指令地址匹配。

BPU（断点单元）

BPU 支持指令读取的断点功能。

MTB（微跟踪缓冲区）

本产品族配有 Cortex-M0+可选组件 MTB，用于支持指令跟踪。

MTB_DWT 控制跟踪的开/停。

SW-DP

本产品族配有 SW-DP，用于支持串行连接协议。

ROM 表

ROM 表的作用是为外部调试工具提供调试组件的地址信息。

单周期 I/O 端口

本产品族配有单周期 I/O 端口，用作对紧密耦合外设的高速访问。

2.1 选项配置

Table2-1 列出了本系列产品 Cortex-M0+内核的选项配置。

Table 2-1 选项配置

功能	TYPE 1-M0+ TYPE 2-M0+ TYPE 3-M0+
中断	32
数据字节端序	小端
SysTick 计时器	有
观察点比较器数量	2
断点比较器数量	4
停止调试支持	有
倍增器	快速型
单周期 I/O 端口	有
唤醒中断控制器	无
向量表偏移寄存器	有
无特权/特权支持	无
存储器保护单元	无
所有寄存器复位	有
指令读取宽度	32-位
MTB	有
调试端口	仅串行线(SW-DP)
串行线多点支持	无

3. 模式

本节描述操作模式的功能。

本产品族可使用以下操作模式：

■ 用户模式
内部 ROM（闪存）启动： CPU 从闪存获得复位向量，然后启动操作。

■ 串行写程序模式
支持闪存串行写入
*：有关本模式的详细信息，参见所用产品的《闪存编程手册》。

操作模式在相应的上电复位、低压检测复位以及 INITX 引脚输入复位释放后确定。

*：有关耗电控制和时钟选择模式的详细信息，参见“低功耗模式”及“时钟”章节。

如何设置操作模式

操作模式通过 MD 引脚（MD0）输入进行设置。

TYPE 1-M0+产品

MD 引脚	操作模式
MD0	
0	用户模式内部 ROM（闪存）启动
1	串行写程序模式

非 TYPE 1-M0+产品

MD 引脚		操作模式
MD1	MD0	
-	0	用户模式内部 ROM（闪存）启动
0	1	串行写程序模式
1	1	禁止设置

启动序列

启动序列的操作模式确定步骤如下：

1. MD 引脚采样
2. 确定操作模式和保留模式数据

各项进程分述如下：

1. MD 引脚采样

通过 MD 引脚（MD0/MD1）输入设置操作模式。 通过上电复位、低压检测复位以及 INITX 引脚输入复位对本项输入进行采样。
在作为采样因子的复位信号释放之前确定 MD0/MD1 引脚输入。

2. 确定操作模式和保留模式数据

直到再次输入其它保持复位采样的输入（MD0/MD1），直到再次输入其它复位信号为止。
由保留的 MD0/MD1 确定操作模式。因此，即使在放出复位信号后修改了 MD0/MD1 也不会影响操作模式。

第 2-1 章：时钟

本章说明操作时钟。

1. 概述
2. 配置
3. 操作
4. 时钟设置步骤示例
5. 寄存器
6. 使用注意事项

代码：9AFCLOCK-C03.0

1. 概述

本节概述时钟生成单元。

时钟生成单元生成操作 MCU 的各类时钟。

本 MCU 的外部和内部振荡时钟统称为源时钟。

源时钟分为以下五种类型：

- 主时钟 (CLKMO)
- 副时钟 (CLKSO)
- 高速 CR 时钟 (CLKHC)
- 低速 CR 时钟 (CLKLC)
- 主 PLL 时钟 (CLKPLL)

选择一类源时钟。本章中，所选择的时钟称为主控时钟。主控时钟是用于操作 MCU 的内部总线时钟源。

将主控时钟分频可生成基本时钟。此外，将基本时钟分频可生成各个总线时钟。

在本章，基本时钟和总线时钟简称为内部总线时钟。以下三类时钟为内部总线时钟：

- 基本时钟 (HCLK/FCLK)
- APB0 总线时钟 (PCLK0)
- APB1 总线时钟 (PCLK1)

除源时钟、主控时钟、内部总线时钟外，还包括以下时钟：

- 用于 USB 的 PLLOUT 时钟（TYPE 3-M0+）
- USB 时钟（TYPE 2-M0+、TYPE 3-M0+）
- CAN 预分频时钟
- 软件看门狗计时器计数时钟

时钟生成单元的功能特性如下：

- 可设置主时钟 (CLKMO) 的振荡稳定等待时间。
- 设置主时钟 (CLKMO) 振荡稳定等待时间结束时生成的中断。
- 可设置副时钟 (CLKSO) 的振荡稳定等待时间。
- 可设置副时钟 (CLKSO) 振荡稳定等待时间结束时生成的中断。
- 可设置主 PLL 时钟 (CLKPLL) 的振荡稳定等待时间。
- 可设置在主 PLL 时钟 (CLKPLL) 的振荡稳定等待时间结束时生成的中断。
- 可设置 PLL 倍频系数。
- 可选择主控时钟。
- 可设置各个内部总线时钟频率的分频比。
- 可选择运行或停止 APB1 总线时钟。
- 可设置软件看门狗计时器计数时钟频率的分频比。
- 可设置软件看门狗计时器计数时钟的运行/停止状态。
- 可设置调试模式下的看门狗计时器计数操作。
- 其中包括使能时钟相关中断、检验中断状态、以及清除中断因素的寄存器。

2. 配置

本节说明时钟生成单元的配置。

源时钟

本 MCU 的外部和内部振荡时钟统称为源时钟。源时钟分为以下五种类型：

■ 主时钟 (CLKMO)

将晶振器等连接至主时钟振荡引脚 (X0, X1) 或使用外部时钟输入生成 CLKMO。

■ 副时钟 (CLKSO)

将晶振器连接至副时钟振荡引脚 (X0A, X1A) 或使用外部时钟输入生成 CLKSO。

■ 高速 CR 时钟 (CLKHC)

CLKHC 是高速 CR 振荡器的输出时钟。

■ 低速 CR 时钟 (CLKLC)

CLKHC 是低速 CR 振荡器的输出时钟。

（注意事项）：低速 CR 时钟是预分频之后的时钟。

有关低速 CR 时钟预分频器，详见“低速 CR 时钟预分频器”一章。

■ 主 PLL 时钟 (CLKPLL)

使用 PLL 时钟倍频电路（PLL 振荡电路）倍增主振荡时钟或高速 CR 时钟生成 CLKPLL。

主控时钟

从源时钟中选择的信号称为主控时钟。主控时钟是所有总线时钟的时钟源。主控时钟值不应大于《数据手册》之“内部操作时钟频率：Fcc（基本时钟 HCLK/FCLK）”中的最大值。

注意事项：若将以下时钟用于主控时钟，参见“E.注释表”中“1.将高速 CR 用于主时钟相关注释”。

- 高速 CR 时钟
- 主 PLL 时钟（将高速 CR 时钟选作 PLL 的输入时钟时）

内部总线时钟

以下信号是内部生成的总线时钟。

■ 基本时钟 (HCLK/FCLK)

HCLK 和 FCLK 统称为基本时钟。 HCLK 和 FCLK 均作为 CPU 的时钟源。
HCLK 是一个与 AHB 总线宏连接的时钟。
时钟频率可设置在主控时钟频率的 1/1 至 1/16 之间。
该时钟在计时器模式、RTC 模式、停止模式、深度待机 RTC 模式、深度待机停止模式下停止运行。
休眠模式下，CPU 停止提供 HCLK，但持续提供 FCLK。

■ APB0 总线时钟 (PCLK0)

PCLK0 是一个外设宏连接到 APB0 总线的时钟。

可将时钟频率设置在基本时钟频率的 1/1 至 1/8 之间。

该时钟在计时器模式、RTC 模式、停止模式、深度待机 RTC 模式、深度待机停止模式下停止运行。

■ APB1 总线时钟 (PCLK1)

PCLK1 是一个外设宏观连接到 APB1 总线的时钟。

可将时钟频率设置在基本时钟频率的 1/1 至 1/8 之间。

该时钟在计时器模式、RTC 模式、停止模式、深度待机 RTC 模式、深度待机停止模式下停止运行。

也可通过设置寄存器来停止时钟源。

源时钟和内部总线时钟之外的时钟

■ 用于 USB 的 PLLOUT 时钟（TYPE 3-M0+）

■ 除了将该时钟用于内部主 PLL 时钟外，带有 USB 的 TYPE 3-M0+产品可将该时钟用作 USB 时钟源。参见通信宏部分中“USB 时钟生成”一章。

■ USB 时钟（TYPE 2-M0+、TYPE 3-M0+）

■ 该时钟是用于 USB 通信的 48MHz 时钟。有关时钟的生成和使用详情，参见通信宏部分中“USB 时钟生成”一章。

■ CAN 预分频时钟

该时钟与 CLKPLL 相同，用于 CAN 预分频器。

时钟分频必须配置在预分频器侧。

该时钟在 RTC 模式、停止模式、深度待机 RTC 模式、深度待机停止模式下停止运行。

有关 CAN 预分频器的操作设置，参见通信宏部分中“CAN 预分频器”一章。

■ 软件看门狗计时器计数时钟 (SWDOGCLK)

SWDOGCLK 是一个连接到 APB0 总线的软件看门狗计时器。

可将时钟频率设置在 APB0 总线时钟频率的 1/1 至 1/8 之间。

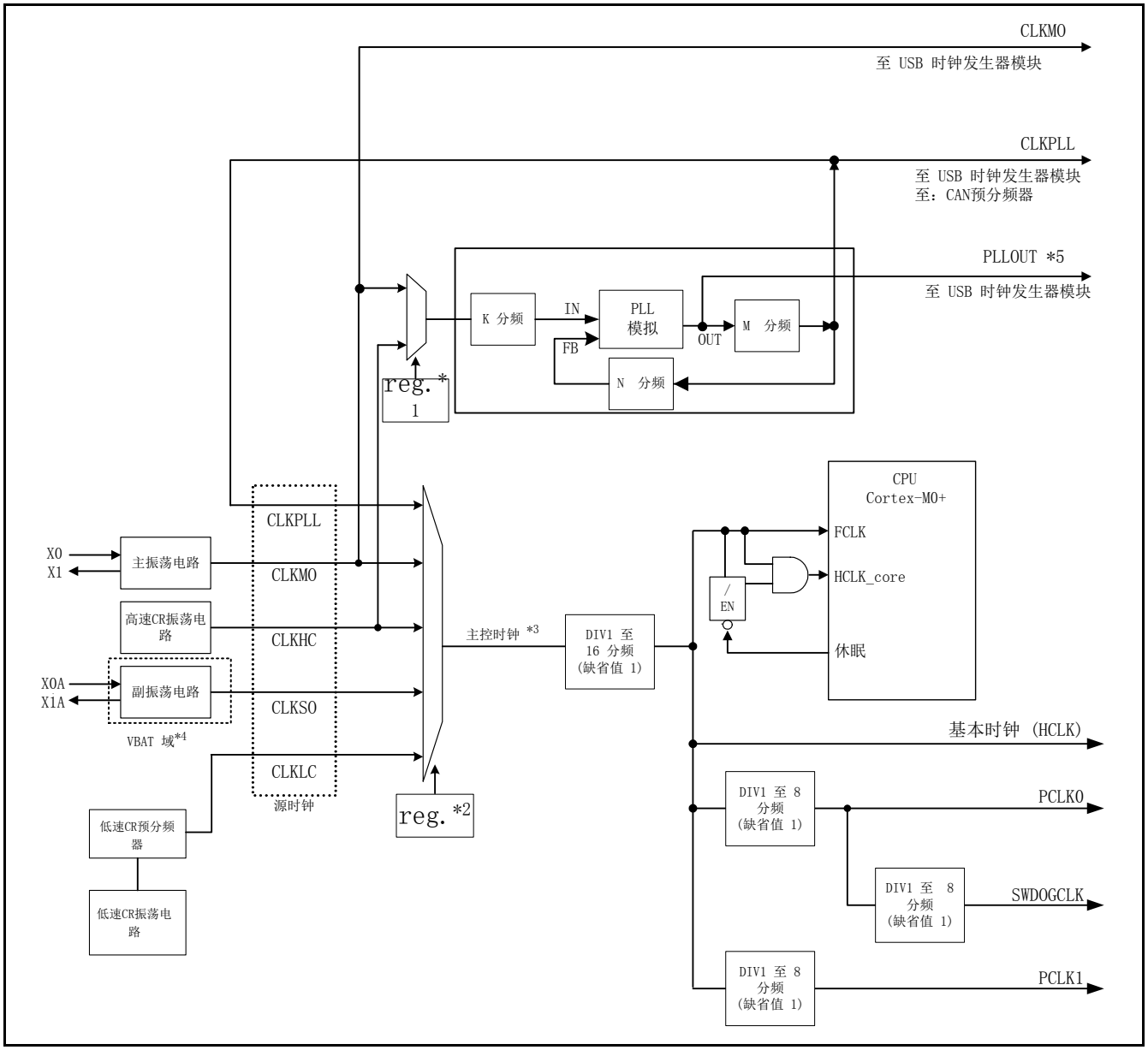
该时钟在计时器模式、RTC 模式、停止模式、深度待机 RTC 模式、深度待机停止模式下停止运行。

有关软件看门狗计时器的操作设置，参见“计时器部分”中“看门狗计时器”一章。

框图

Figure 2-1 所示为时钟生成单元框图。

Figure 2-1 时钟生成单元框图



*1: PSW_TMR: PINC（PLL 输入时钟选择位）

*2: SCM_CTL: RCS[2:0]（主控时钟开关控制位）

*3: 主控时钟频率不应大于基本时钟 (HCLK/FCLK) 的最大频率。有关基本时钟 (HCLK/FCLK) 的最大频率，参见所使用产品的《数据手册》。

*4: TYPE 2-M0+产品中，副振荡电路位于不同的独立功率域中。详细内容参见“VBAT 域”一章。

*5: TYPE 3-M0+ 产品中，将 PLLOUT 时钟用作 USB 时钟的源时钟。

3. 操作

本节说明时钟生成单元。

3.1 选择时钟模式

时钟模式的定义（选择主控时钟）

MCU 时钟模式由系统时钟模式控制寄存器选择的源时钟定义。有五类时钟模式：主时钟模式、副时钟模式、高速 CR 时钟模式、低速 CR 时钟模式和主 PLL 时钟模式。

- 主时钟模式

主时钟模式中，将主时钟 (CLKMO) 用作主控时钟。该时钟运行用于操作 CPU 及大多数外设功能的总线时钟。

PLL 时钟 (CLKPLL) 的状态取决于系统时钟模式控制寄存器 (SCM_CTL) 中 PLLE 位的设置；副时钟 (CLKSO) 的状态则取决于系统时钟模式控制寄存器 (SCM_CTL) 中 SOSCE 位的设置；而高速 CR 时钟(CLKHC)的状态取决于系统时钟模式控制寄存器 (SCM_CTL) 中 HCRE 位的设置。此外，CSV_CTL 的 MCSVE/FCSDE 也激活高速 CR 时钟 (CLKHC)。低速 CR 时钟 (CLKLC) 不能通过用户程序停止。

- 副时钟模式

副时钟模式中，将副时钟(CLKSO)用作主控时钟。该时钟运行用于操作 CPU 及大多数外设功能的总线时钟。

主时钟 (CLKMO)、高速 CR 时钟 (CLKHC) 及主 PLL 时钟 (CLKPLL)通过硬件停止。低速 CR 时钟 (CLKLC)不能通过用户程序停止。

- 高速 CR 时钟模式

高速 CR 时钟模式中，将高速 CR 时钟 (CLKHC) 用作主控时钟。该时钟运行用于操作 CPU 及大多数外设功能的总线时钟。

主时钟 (CLKMO)、主 PLL 时钟 (CLKPLL) 和副时钟 (CLKSO) 的状态具体取决于系统时钟模式控制寄存器 (SCM_CTL) 中 MOSCE、PLLE 和 SOSCE 位的设置。高速 CR 时钟 (CLKHC) 和低速 CR 时钟 (CLKLC) 不能通过用户程序停止。

- 低速 CR 时钟模式

低速 CR 时钟模式中，将低速 CR 时钟 (CLKHC) 用作主控时钟。该时钟运行用于操作 CPU 及大多数外设功能的总线时钟。

低速 CR 时钟模式中，主时钟 (CLKMO)、高速 CR 时钟 (CLKHC) 和主 PLL 时钟 (CLKPLL) 通过硬件停止副时钟 (CLKSO) 的状态具体取决于系统时钟模式控制寄存器 (SCM_CTL) 中 SOSCE 位的设置。

- 主 PLL 时钟模式

主 PLL 时钟模式中，将主 PLL 时钟 (CLKPLL) 用作主控时钟。该时钟运行用于操作 CPU 及大多数外设功能的总线时钟。

副时钟 (CLKSO) 的状态具体取决于系统时钟模式控制寄存器 (SCM_CTL) 中 SOSCE 位的设置。高速 CR 时钟 (CLKHC) 的状态具体取决于某些寄存器的设置，详见“低功耗模式”一章。主时钟 (CLKMO) 状态具体取决于 PSW_TMR 内 PINC 或 SCM_CTL 内 MOSCE 的设置。低速 CR 时钟 (CLKLC) 不能通过用户程序停止。

3.2 内部总线时钟分频控制

本节说明内部总线时钟分频。

可单独设置各个内部总线时钟的基本时钟分频比。

本功能还可针对各电路设置优化工作频率。

不同产品的内部总线时钟最大频率各不相同。详见所用产品的《数据手册》。

设置内部总线时钟的分频比时，要使用到基本时钟预分频寄存器 (BSC_PSR)、APB0 预分频寄存器 (APBC0_PSR) 和 APB1 预分频寄存器 (APBC1_PSR) 。有关各寄存器详见 5.寄存器。

设置总线时钟分频

- 设置的分频比不是通过软件复位清除。在软件复位前保持最新值。
- 通过非软件复位以外的复位对分频比值进行初始化。
- 更改主控时钟初始值为更快的源时钟之前，确保设置分频比。
- 若主控时钟、PLL 倍频和分频比设置的合并值超过各内部总线的最大工作频率，将不能保证执行与设置相对应的操作。

3.3 PLL 时钟控制

本节说明 PLL 时钟控制。

PLL 时钟控制电路用于从主时钟或高速 CR 时钟生成主 PLL 时钟。PLL 振荡电路能用于使能/禁用操作（振荡）、选择输入时钟、设置稳定等待时间以及设置倍频。

PLL 操作

主 PLL 时钟的操作说明如下：

- 用 PLL 时钟振荡稳定等待时间设置寄存器 (PSW_TMR) 配置以下设置：
- 选择 PLL 输入时钟
- 设置主 PLL 时钟稳定等待时间
- 必须使能系统时钟模式控制寄存器 (SCM_CTL) 的 PLL 振荡使能位 (PLLE)，使 PLL 电路起振。
- 当 PLL 时钟稳定等待时间结束且系统时钟模式状态寄存器 (SCM_STR) 的 PLL 振荡稳定位指示状态已稳定时，转换至主 PLL 时钟模式的准备即完成。
- 必须将系统时钟模式控制寄存器 (SCM_CTL) 的主控时钟开关控制位 (RCS[2:0]) 设置为主 PLL 时钟模式 (RCS[2:0]=010)，以转换至主 PLL 时钟模式。

设置主 PLL 时钟振荡稳定等待时间

详见 5.8 PLL 时钟稳定等待时间设置寄存器 (PSW_TMR)

注意事项：

- *PLL 时钟控制电路的框图参见 2.配置*
- *有关各内部总线时钟的分频设置顺序，参见 4 时钟设置步骤示例*
- *有关振荡稳定等待时间，参见 3.4 振荡稳定等待时间*
- *有关 PLL 的输入时钟内高速 CR 的选择，参见附录 E. 注释表中"1.将高速 CR 用于主控时钟相关注释"。*

设置生成主 PLL 时钟的倍频系数

必须使用 PLL 控制寄存器 1 (PLL_CTL1) 和 PLL 控制寄存器 2 (PLL_CTL2) 设置 PLL 倍频电路的各分频时钟。分频设置示例参见 Table 3-1 和 Table 3-2。

Table 3-1PLL 倍频系数设置示例（TYPE 1-M0+、TYPE 2-M0+)

输入时钟	K	PLLin	N	PLLout	M	CLKPLL
4 MHz	1	4 MHz	2	80 MHz	10	8 MHz
4 MHz	1	4 MHz	4	80 MHz	5	16 MHz
4 MHz	1	4 MHz	5	80 MHz	4	20 MHz
4 MHz	1	4 MHz	6	120 MHz	5	24 MHz
4 MHz	1	4 MHz	9	108 MHz	3	36 MHz
4 MHz	1	4 MHz	10	80 MHz	2	40 MHz
8 MHz	1	8 MHz	5	80 MHz	2	40 MHz
8 MHz	2	4 MHz	10	80 MHz	2	40 MHz
12 MHz	3	4 MHz	10	80 MHz	2	40 MHz
16 MHz	2	8 MHz	5	80 MHz	2	40 MHz
16 MHz	4	4 MHz	10	80 MHz	2	40 MHz
24 MHz	3	8 MHz	5	80 MHz	2	40 MHz

Table 3-2 PLL 倍频系数设置示例（TYPE 3-M0+）

输入时钟	K	PLLin	N	PLLout	M	CLKPLL
8 MHz	1	8 MHz	2	80 MHz	5	16 MHz
8 MHz	1	8 MHz	3	96 MHz	4	24 MHz
8 MHz	1	8 MHz	4	96 MHz	3	32 MHz
8 MHz	1	8 MHz	5	80 MHz	2	40 MHz
12 MHz	1	12 MHz	3	144 MHz	4	36 MHz
16 MHz	2	8 MHz	5	80 MHz	2	40 MHz
24 MHz	3	8 MHz	5	80 MHz	2	40 MHz
48MHz	6	8MHz	5	80MHz	2	40MHz

注意事项：

- *PLL 特性参见所用产品的《数据手册》。*
- *PLLin 设置为《数据手册》所述 PLL 输入时钟的频率：f_{PLL} 的值范围内。*
- *M×N 值为 PLLin 的倍频系数。将此值设置在《数据手册》中"PLL 倍率"所示范围内。*
- *PLLin 的频率乘以 M×N 得到 PLLout。将此值设置为《数据手册》中 PLL 宏振荡时钟频率：f_{PLLO} 所示范围内。*
- *PLLout 的值除以 M 得到 CLKPLL。*
- *有关 PLL 及分频器的配置参见 Figure 2-1。*
- *主控时钟/CLKPLL 值不应大于《数据手册》之"内部操作时钟频率：Fcc（基本时钟 HCLK/FCLK）"中的最大值。*
- *TYPE 3-M0+ 产品中，PLLout 时钟可用于生成 USB 时钟的源时钟。详见 USB 时钟生成章节。*

3.4 振荡稳定等待时间

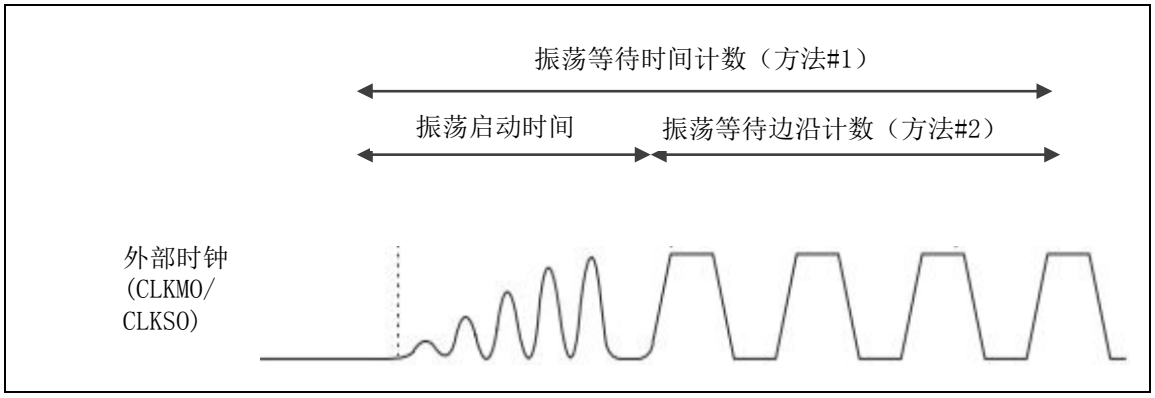
本节说明振荡稳定等待时间。

若源时钟未处于稳定工作状态，需要有振荡稳定等待时间。在振荡稳定等待时间内，停止提供内部时钟和外部时钟。有两种方法等待至稳定等待时间结束，即在时钟稳定等待时间寄存器 (CSW_TMR) 或 PLL 时钟振荡稳定等待时间设置寄存器 (PSW_TMR) 内设定时间值。等待时间结束后，对应的振荡器准备工作，可将时钟用作主控时钟。

时钟稳定等待计数方法

- 有两种计数外部时钟 (CLKMO/CLKSO) 稳定时间的方法。
  1. 一种方法是通过内部 CR 时钟 (CLKHC/CLKLC) 计数外部时钟 (CLKMO/CLKSO) 的稳定时间。使用这种方法时，超出特定时间后外部时钟 (CLKMO/CLKSO) 稳定。
  2. 另一种方法是通过计数外部时钟 (CLKMO/CLKSO) 正沿的数量进行其自身的等待计数。
- 每种方法中，都需要选择假定外部时间稳定必须计数的时长/时钟沿数量。
- 时钟稳定时间/时钟沿数量选择示例如下所示。

Figure 3-1 振荡等待计数方法



- 通过使能或禁用时钟监视器选择时钟稳定方法。

  CSV_CTL.FCSDE=0 和 CSV_CTL.MCSVE=0: 选择 CLKMO 时钟沿数量计数方法 2。

  CSV_CTL.FCSDE=1 或 CSV_CTL.MCSVE=1: 选择 CLKHC 计数时间方法 1。

  CSV_CTL.SCSVE=0: 选择 CLKSO 时钟沿数量计数方法 2。

  CSV_CTL.SCSVE=1: 选择 CLKLC 计数时间方法 1。

振荡稳定等待时间优先序列

若某些时钟通过模式转换起振，时钟控制器将根据设定的优先顺序计算各时钟的振荡稳定等待时间。

- 转换至主时钟模式
  - 若使能 MCSV 或 FCS：
    低速 CR ->副 OSC ->高速 CR ->PLL（高速 CR 输入） ->主 OSC-> PLL（主 OSC 输入）
  - 若禁用 MCSV 或 FCS：
    低速 CR ->副 OSC ->主 OSC ->PLL（主 OSC 输入）->高速 CR-> PLL（高速 CR 输入）
- 转换至副时钟模式
  低速 CR ->副 OSC
- 转换至高速 CR 时钟模式
  低速 CR ->副 OSC ->高速 CR ->PLL（高速 CR 输入） ->主 OSC-> PLL（主 OSC 输入）
- 转换至低速 CR 时钟模式
  低速 CR ->副 OSC
- 转换至主时钟 PLL 模式
  - 若使能 MCSV 或 FCS：
    低速 CR ->副 OSC ->高速 CR ->主 OSC -> PLL（主 OSC 输入）
  - 若禁用 MCSV 或 FCS：
    低速 CR ->副 OSC ->主 OSC ->PLL（主 OSC 输入）->高速 CR
- 转换至高速时钟 PLL 模式
  低速 CR ->副 OSC ->高速 CR ->PLL（高速 CR 输入）->主 OSC

设置振荡稳定等待时间

- 主时钟 (CLKMO)
  用时钟稳定等待时间寄存器 (CSW_TMR) 设置主时钟的稳定等待时间。通过 CLKHC（若使能 CLKMO 的时钟监视器）/CLKMO（若禁用 CLKMO 的时钟监视器）计数设定时间值。
- 副时钟 (CLKSO)
  用时钟稳定等待时间寄存器 (CSW_TMR) 设置副时钟的稳定等待时间。通过 CLKLC（若使能 CLKSO 的 CSV）/CLKSO（若禁用 CLKSO 的 CSV）计数设定时间值。
- 主 PLL 时钟
  用 PLL 时钟振荡稳定等待时间设置寄存器 (PSW_TMR) 配置以下设置：使用 CLKPLL 计数设定的时间值。
- 选择 PLL 输入时钟
- 设置主 PLL 时钟稳定等待时间

等待振荡稳定性的原因

- 通过软件使能振荡后：
  若将系统时钟模式控制寄存器 (SCM_CTL) 内的 PLLE、SOSCE、MOSCE 和 HCRE 位设置为 "1"，各相关振荡器将在振荡稳定等待时间内等待。
- 从 RTC 模式唤醒至计时计数器中断、RTC 中断、外部中断时：
  通过计时计数器中断、RTC 中断、外部中断唤醒至 RTC 模式前的时钟模式。源时钟的硬件自动等待振荡稳定等待时间。

■ 通过外部中断从停止模式唤醒时：
通过外部时钟唤醒至时钟模式状态，即停止模式前的状态。停止模式时，所有源时钟停止，因此硬件会在振荡稳定等待时间内自动等待。

■ 使能 PLL 操作后：
使能 PLL 操作后，要在 PLL 振荡稳定等待时间内等待。

注意事项：

- *振荡稳定等待时间各设定值必须在使能时钟前更改*
- *软件复位后，振荡稳定等待时间不适用。*
- *计数完成后将激活振荡稳定等待完成标志。振荡稳定等待时间的设置如果太短，振荡稳定等待时间可能在振荡器稳定前完成。*
- *由于主时钟和副时钟振荡器的稳定等待时间取决于振荡器类型（晶体式、陶瓷式等），因此必须选择所用振荡器恰当的振荡稳定等待时间。*
- *参考所用产品的《数据手册》所述电气特性中的 PLL 时钟 LOCKUP 时间，设置 PLL 振荡稳定等待时间。*

3.5 中断因素

本节说明时钟相关中断因素。

时钟生成单元具有以下中断因素：

中断因素

时钟生成单元有以下四类中断因素：

- FCS（异常频率检测）中断

  若使能 FCS（异常频率检测）且检测到主时钟异常频率，将发生中断。

- 主 PLL 时钟振荡稳定等待完成中断

  主 PLL 时钟振荡稳定等待时间结束时发生中断。

- 副时钟振荡稳定等待完成中断

  副时钟振荡稳定等待时间结束时发生中断。

- 主时钟振荡稳定等待完成中断

  主时钟振荡稳定等待时间结束时发生中断。

寄存器

各中断因素有以下三类寄存器：

- 中断使能寄存器(INT_ENR)

  该寄存器使能各中断。

- 中断状态寄存器(INT_STR)

  该寄存器指示各中断状态。该寄存器是只读寄存器。

- 中断清除寄存器(INT_CLR)

  该寄存器清除各个中断因素。该寄存器是只写寄存器。

4. 时钟设置步骤示例

4.1 设置步骤示例（TYPE 1-M0+、TYPE 3-M0+）

本节说明 TYPE 1-M0+,TYPE 3-M0+产品的时钟设置步骤示例。

Figure 4-1 时钟设置步骤示例（上电 ->高速 CR 运行模式 ->期望时钟模式）

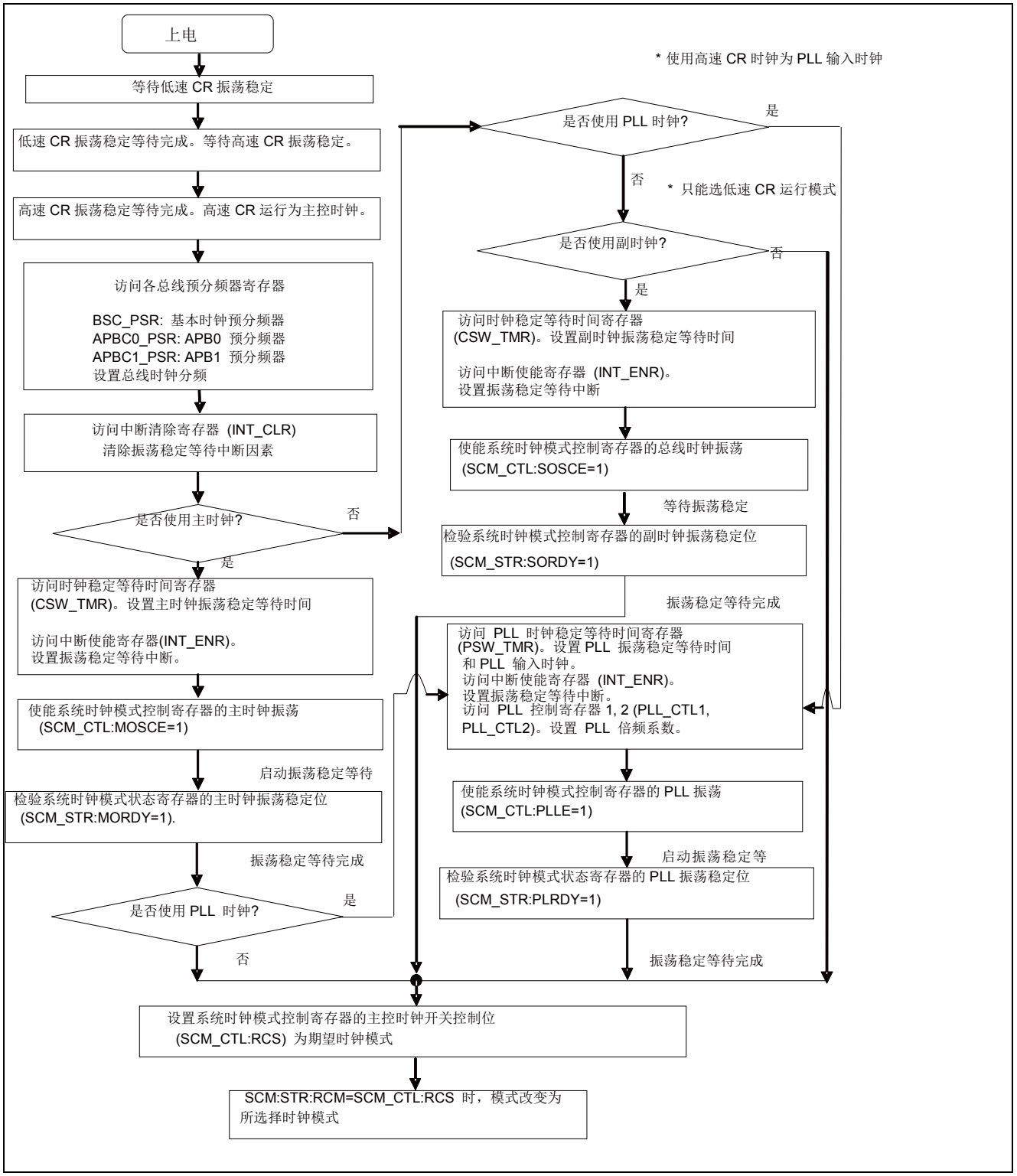
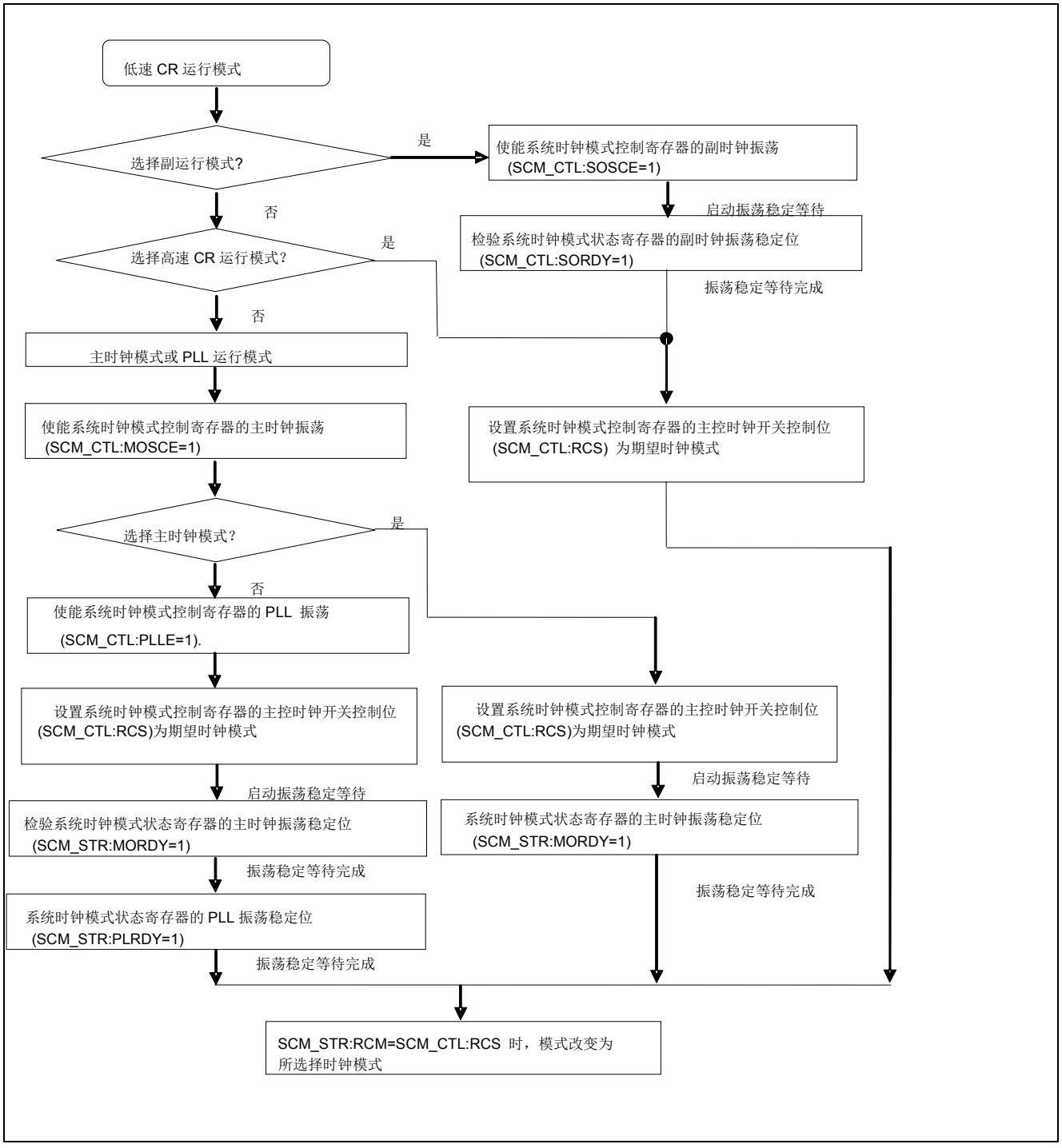


Figure4-2 时钟设置步骤示例（低速 CR 运行模式->期望时钟运行模式）



注意事项：

- *Figure 4-2 假设已预先配置各时钟的振荡稳定等待时间、中断、PLL 倍频系数和总线时钟分频设置，所以流程图中省略上述内容。*

- *副时钟模式/低速 CR 时钟模式时，通过硬件停止主时钟 (CLKMO)、高速 CR (CLKHC) 和主 PLL 时钟 (CLKPLL)。因此 CLKMO/CLKHC/CLKPLL 不会起振，只将振荡使能位设置为 1。更改 SCM_CTL:RCS 位，将振荡使能位设置为 1，才会起振。*
- *若主时钟/副时钟振荡稳定等待时间过短，且振荡稳定等待时间在振荡器稳定之前就结束，可通过时钟监视功能进行复位。*

4.2 设置步骤示例（TYPE 2-M0+）

本节说明 TYPE 2-M0+ 产品的时钟设置步骤示例。

Figure4-3 时钟设置步骤示例（上电->高速 CR 运行模式->期望时钟模式）

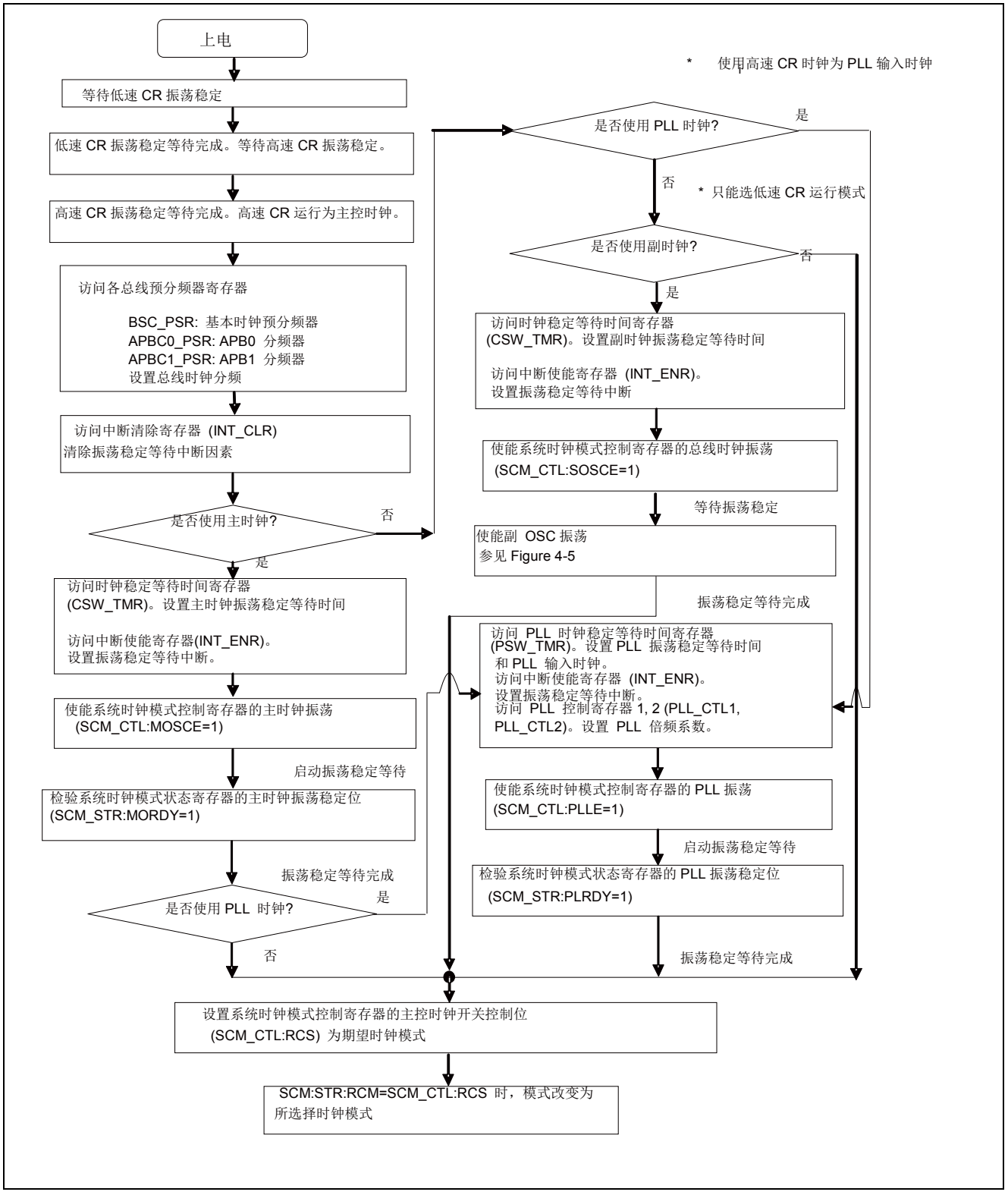
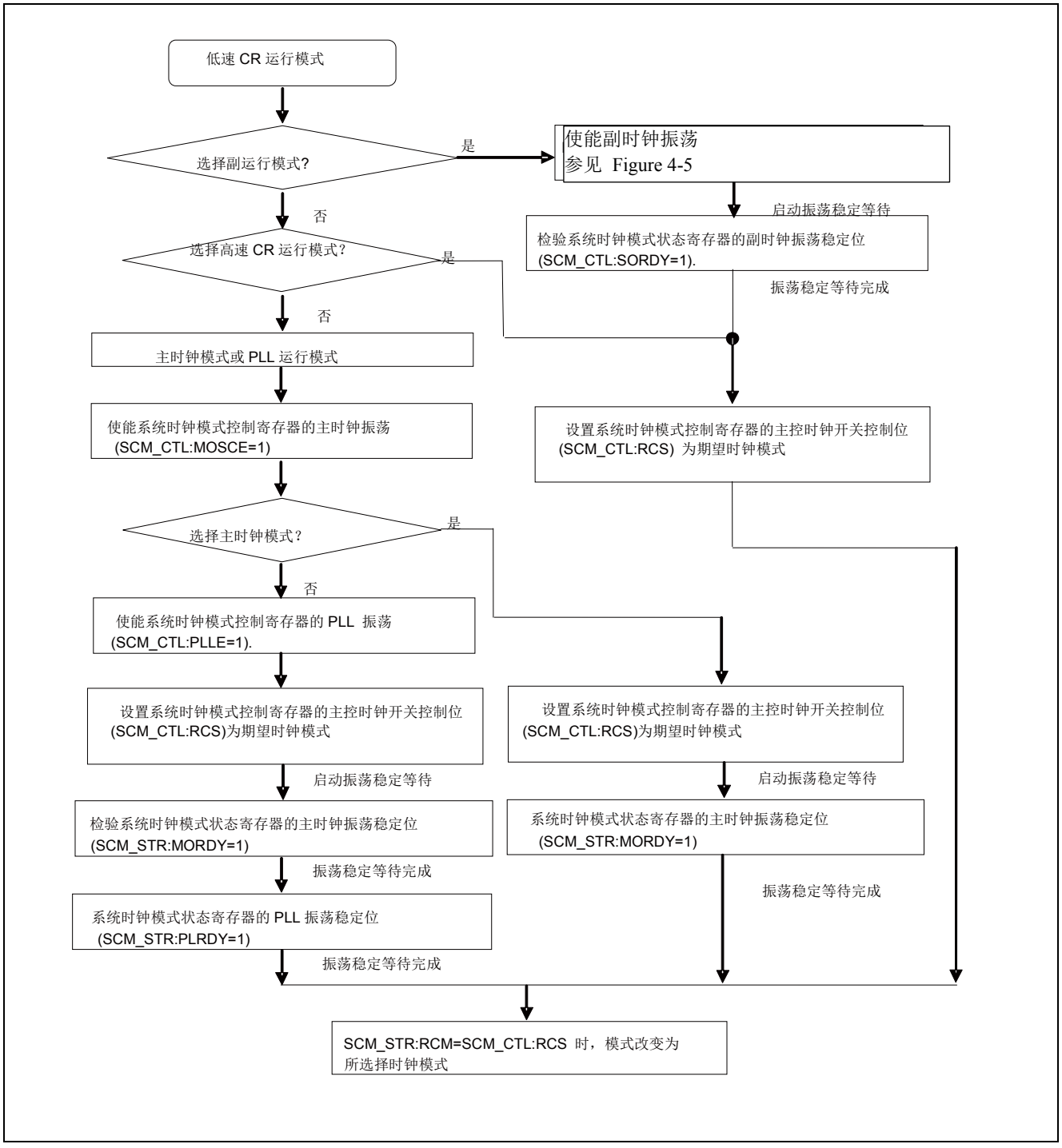


Figure4-4 时钟设置步骤示例（低速 CR 运行模式->期望时钟模式）

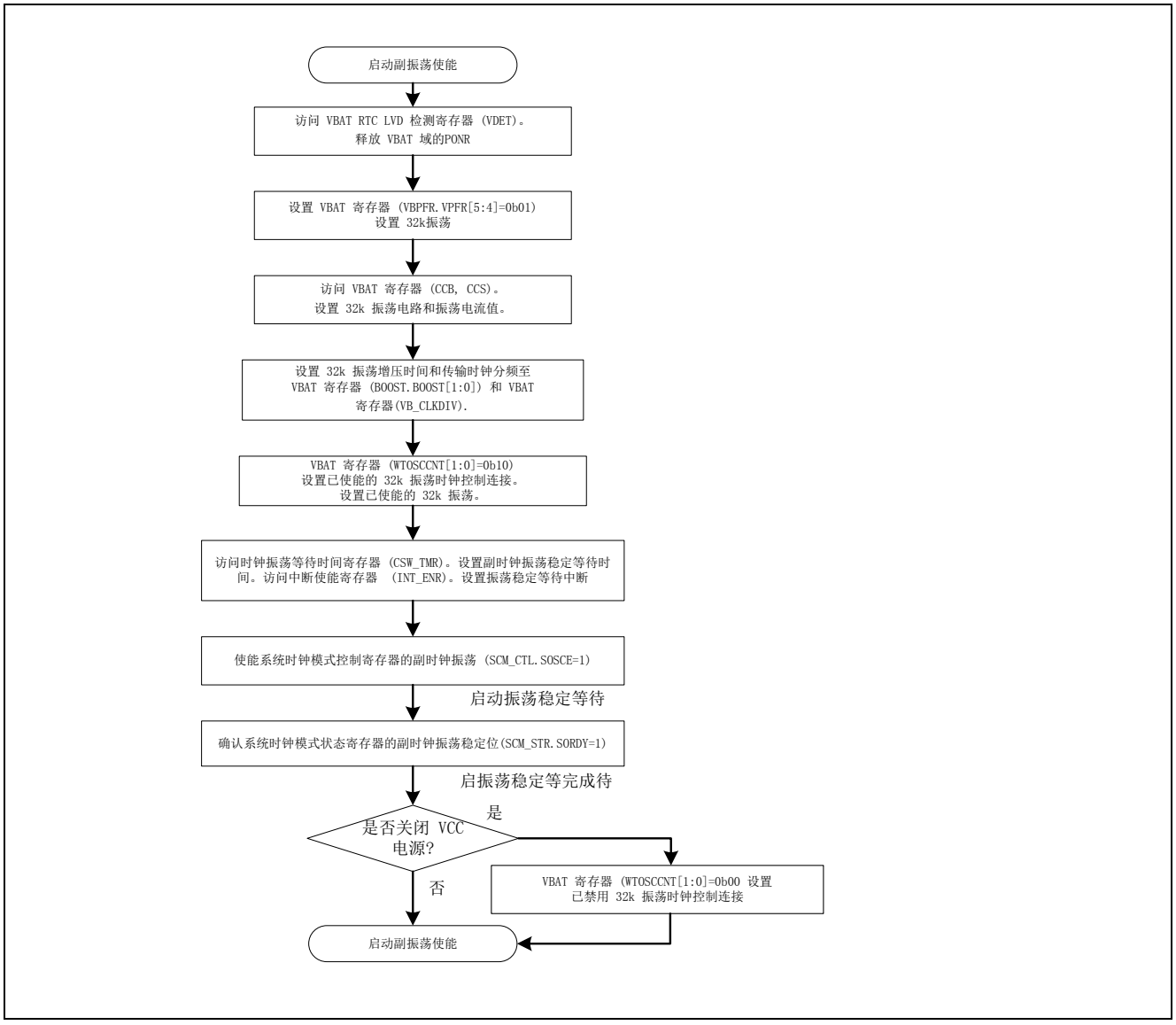


注意事项：

- *Figure 4-4 假设已预先配置各时钟的振荡稳定等待时间、中断、PLL 倍频系数和总线时钟分频设置，所以流程图中省略上述内容。*

- *副时钟模式/低速 CR 时钟模式时，通过硬件停止主时钟 (CLKMO)、高速 CR (CLKHC) 和主 PLL 时钟 (CLKPLL)。因此 CLKMO/CLKHC/CLKPLL 不会起振，只将振荡使能位设置为 1。更改 SCM_CTL:RCS 位，将振荡使能位设置为 1，才会起振。*
- *若主时钟/副时钟振荡稳定等待时间过短，且振荡稳定等待时间在振荡器稳定之前就结束，可通过时钟监视功能进行复位。*

Figure 4-5 副振荡设置步骤示例



注意事项：

- *设置的副时钟稳定等待时间 (SCM_CTL.SOWT) 大于 VBAT RTC 32 k 振荡增压设置时间 [BOOST:BOOST[1:0]]。*
- *禁用以下设置组合：*
  - *若禁用 VBAT 寄存器的 32k 振荡时钟控制链路 (WTOSCCNT.SOSCNTL=0)，且禁用 32k 振荡 (WTOSCCNT.SOSCEX=1)，禁止已使能系统时钟模式控制寄存器的副时钟模式振荡*

(SCM_CTL.SOSCE=1) 和已使能 CSV 控制寄存器的副 CSV 功能 (CSV_CTL.SVSVE=1) 的设置组合。

- *有关 VBAT RTC 详细内容，参见 VBAT 域一章。*

5. 寄存器

本节说明时钟生成单元的寄存器。

时钟生成单元寄存器列表

Table5-1 时钟生成单元寄存器列表

缩写	寄存器名称	参考章节
SCM_CTL	系统时钟模式控制寄存器	5.1
SCM_STR	系统时钟模式状态寄存器	5.2
BSC_PSR	基本时钟预分频寄存器	5.3
APBC0_PSR	APB0 预分频寄存器	5.4
APBC1_PSR	APB1 预分频寄存器	5.5
SWC_PSR	软件看门狗时钟预分频寄存器	5.6
CSW_TMR	时钟稳定等待时间寄存器	5.7
PSW_TMR	PLL 时钟稳定等待时间设置寄存器	5.8
PLL_CTL1	PLL 控制寄存器 1	5.9
PLL_CTL2	PLL 控制寄存器 2	5.10
DBWDT_CTL	调试断点看门狗计时器控制寄存器	5.11
INT_ENR	中断使能寄存器	5.12
INT_STR	中断状态寄存器	5.13
INT_CLR	中断清除寄存器	5.14

5.1 系统时钟模式控制寄存器 (SCM_CTL)

系统时钟模式控制寄存器 (SCM_CTL) 选择主时钟，并使能/禁用时钟振荡。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	RCS[2:0]			PLLE	SOSCE	保留	MOSCE	HCRE
属性		R/W		R/W	R/W	-	R/W	R/W
初始值		000		0	0	-	0	1

寄存器功能

[bit7:5] RCS[2:0]: 主控时钟开关控制位

bit7	bit6	bit5	描述
0	0	0	高速 CR 时钟 [初始值]
0	0	1	主时钟
0	1	0	主 PLL 时钟
0	1	1	禁止设置
1	0	0	低速 CR 时钟
1	0	1	副时钟
1	1	0	禁止设置
1	1	1	禁止设置

[bit4] PLLE: PLL 振荡使能位

位	描述
0	禁用 PLL 振荡 [初始值]
1	使能 PLL 震动

[bit3] SOSCE: 副时钟振荡使能位

位	描述
0	禁用副时钟振荡 [初始值]
1	使能副时钟振荡

[bit2] 保留：保留位

读取值未定义。这些位写入无效。

[bit1] MOSCE: 主时钟振荡使能位

位	描述
0	禁用主时钟振荡 [初始值]
1	使能主时钟振荡

[bit0] HCRE: 高速 CR 时钟振荡使能位

位	描述
0	禁用高速 CR 振荡
1	使能高速 CR 振荡【初始值】

注意事项：

- *软件复位不能初始化本寄存器。*
- *改变时钟模式时，使能位设置为期望时钟振荡的转换。然后可更改时钟开关控制位 (SCM_CTL:RCS[2:0])。*
- *若 RTCE 位 (PMD_CTL:RTCE) 为 1，无论 SOSCE 位和 SORDY 位的值如何，均为副时钟振荡使能状态。*
- *仅在 SORDY 位为 1 时，才能使能 RTCE 位 (PMD_CTL:RTCE)写入 1。*
- *不具有 RTC 模式和深度待机 RTC 模式的产品中没有 RTCE 位 (PMD_CTL:RTCE)。参见低功耗模式一章中 Table 1-1。*
- *若 FCSDE 位 (CSV_CTL:FCSDE) 或 MCSVE 位 (CSV_CTL:MCSVE) 为 "1"，无论 HCRE 位的值如何，均为高速 CR 时钟振荡使能状态。*
- *TYPE 2-M0+ 产品中，首先在开启电源后要求设置 VBAT RTC 寄存器，使能副时钟振荡。有关使能副时钟振荡，参见 Figure 4-5。*
- *若有时钟处于稳定等待状态，不得使能其它时钟振荡。*

5.2 系统时钟模式状态寄存器 (SCM_STR)

系统时钟模式状态寄存器 (SCM_STR) 指示选择用为主控时钟的时钟以及时钟振荡稳定性的等待状态。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	RCM[2:0]			PLRDY	SORDY	保留	MORDY	HCRDY
属性		R		R	R	-	R	R
初始值		000		0	0	-	0	1

寄存器功能

[bit7:5] RCM[2:0]: 主控时钟选择位

bit7	bit6	bit5	描述
0	0	0	高速 CR 时钟 [初始值]
0	0	1	主时钟
0	1	0	主 PLL 时钟
0	1	1	禁止设置
1	0	0	低速 CR 时钟
1	0	1	副时钟
1	1	0	禁止设置
1	1	1	禁止设置

[bit4] PLRDY: PLL 振荡稳定位

位	描述
0	处于稳定等待或振荡停止状态 [初始值]
1	处于稳定状态

[bit3] SORDY: 副时钟振荡稳定位

位	描述
0	处于稳定等待或振荡停止状态 [初始值]
1	处于稳定状态

[bit2] 保留：保留位

读取值未定义。这些位写入无效。

[bit1] MORDY: 主时钟振荡稳定位

位	描述
0	处于稳定等待或振荡停止状态 [初始值]
1	处于稳定状态

[bit0] HCRDY: 高速 CR 时钟振荡稳定位

位	描述
0	处于稳定等待或振荡停止状态
1	处于稳定状态 [初始值]

注意事项：

- *软件复位不能初始化本寄存器。*
- *若 RTCE 位 (PMD_CTL:RTCE) 为 1，无论 SOSCE 位和 SORDY 位的值如何，均为副时钟振荡使能状态。*
- *仅在 SORDY 位为 1 时，才能使能写入 "1" 至 RTCE 位 (PMD_CTL:RTCE)。*

5.3 基本时钟预分频寄存器 (BSC_PSR)

基本时钟预分频寄存器 (BSC_PSR) 设置基本时钟的分频比。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	保留						BSR	
属性			-				R/W	
初始值			-				000	

寄存器功能

[bit7:3] 保留：保留位

这些位的读取值为 0b00000。写入时将这些位设置为 0b00000。

[bit2:0] BSR: 基本时钟分频比设置位

bit2	bit1	bit0	描述
0	0	0	1/1 [初始值]
0	0	1	1/2
0	1	0	1/3
0	1	1	1/4
1	0	0	1/6
1	0	1	1/8
1	1	0	1/16
1	1	1	禁止设置

注意事项：

- *软件复位不能初始化本寄存器。*

5.4 APB0 预分频寄存器 (APBC0_PSR)

APB0 预分频寄存器 (APBC0_PSR) 设置 APB0 总线时钟分频。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	保留						APBC0	
属性	-						R/W	
初始值	-						00	

寄存器功能

[bit7:2] 保留：保留位

这些位的读取值为 0b000000。写入时将这些位设置为 0b000000。

[bit1:0] APBC0: APB0 总线时钟分频设置位

bit1	bit0	描述
0	0	1/1 [初始值]
0	1	1/2
1	0	1/4
1	1	1/8

注意事项：

- *软件复位不能初始化本寄存器。*

5.5 APB1 预分频寄存器 (APBC1_PSR)

APB1 预分频寄存器 (APBC1_PSR) 设置 APB1 总线时钟分频。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	APBC1EN	保留		APBC1RST	保留		APBC1	
属性	R/W	-		R/W	-		R/W	
初始值	1	-		0	-		00	

寄存器功能

[bit7] APBC1EN: APB1 时钟使能位

位	描述
0	禁用 PCLK1 输出
1	使能 PCLK1 输出 [初始值]

[bit6:5] 保留：保留位

这些位的读取值为 0b00。写入时将这些位设置为 0b00。

[bit4] APBC1RST: APB1 总线复位控制位

位	描述
0	APB1 总线复位，无效 [初始值]
1	APB1 总线复位，有效

[bit3:2] 保留：保留位

这些位的读取值为 0b00。写入时将这些位设置为 0b00。

[bit1:0] APBC1: APB1 总线时钟分频设置位

bit1	bit0	描述
0	0	1/1 [初始值]
0	1	1/2
1	0	1/4
1	1	1/8

注意事项：

- *软件复位不能初始化本寄存器。*

5.6 软件看门狗时钟预分频寄存器 (SWC_PSR)

软件看门狗时钟预分频寄存器 (SWC_PSR) 设置分频，并使能软件看门狗时钟的输出。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	TESTB	保留					SWDS	
属性	R/W	-					R/W	
初始值	X	-					00	

寄存器功能

[bit7] TESTB: TEST 位

位	描述
0	禁止设置
1	总写入 "1"。

注意事项：此位的读取值未定义。

[bit6:2] 保留：保留位

这些位的读取值为 0b00000。写入时将这些位设置为 0b00000。

[bit1:0] SWDS: 软件看门狗时钟分频比设置位

bit1	bit0	描述
0	0	设置 PCLK0 的分频比为 1/1 [初始值]
0	1	设置 PCLK0 的分频比为 1/2
1	0	设置 PCLK0 的分频比为 1/4
1	1	设置 PCLK0 的分频比为 1/8

注意事项：

- *软件复位不能初始化本寄存器。*
- *该寄存器写入值时，确认将 TESTB 位设置为 "1" 。*

5.7 时钟稳定等待时间寄存器(CSW_TMR)

时钟稳定等待时间寄存器 (CSW_TMR) 设置主/副时钟的稳定等待时间。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	SOWT				MOWT			
属性	R/W				R/W			
初始值	0000				0000			

寄存器功能

[bit7:4] SOWT: 副时钟稳定等待时间设置位

通过 CLKLC 或 CLKSO 计数稳定等待时间。

bit7	bit6	bit5	bit4	描述	计算示例 1 CLKLC=100kHz	计算示例 2 CLKSO=32.768kHz
0	0	0	0	2^{10} 周期[初始值]	大约 10.3 ms	大约 31.3 ms
0	0	0	1	2^{11} 周期	大约 20.5 ms	大约 62.5 ms
0	0	1	0	2^{12} 周期	大约 41 ms	大约 125 ms
0	0	1	1	2^{13} 周期	大约 82 ms	大约 250 ms
0	1	0	0	2^{14} 周期	大约 164 ms	大约 500 ms
0	1	0	1	2^{15} 周期	大约 327 ms	大约 1.00 s
0	1	1	0	2^{16} 周期	大约 655 ms	大约 2.00 s
0	1	1	1	2^{17} 周期	大约 1.31 s	大约 4.00 s
1	0	0	0	2^{18} 周期	大约 2.62 s	大约 8.00 s
1	0	0	1	2^{19} 周期	大约 5.24 s	大约 16.0 s
1	0	1	0	2^{20} 周期	大约 10.49 s	大约 32.0 s
1	0	1	1	2^{21} 周期	大约 20.97 s	大约 64.0 s
1	1	0	0	2^{1} 周期	大约 0.02 ms	大约 0.06 ms
1	1	0	1	2^{2} 周期	大约 0.04 ms	大约 0.12 ms
1	1	1	0	2^{3} 周期	大约 0.08 ms	大约 0.24 ms
1	1	1	1	2^{4} 周期	大约 0.16ms	大约 0.49 ms

[bit3:0] MOWT: 主时钟稳定等待时间设置位

通过 CLKHC 或 CLKMO 计数稳定等待时间。

bit3	bit2	bit1	bit0	描述	计算示例		
					CLKHC=4MHz 或 CLKMO=4MHz	**CLKHC=8MHz 或 CLKMO=8MHz**	CLKMO=40MHz
0	0	0	0	2^1 周期[初始值]	大约 500 ns	大约 250 ns	大约 50 ns
0	0	0	1	2^5 周期	大约 8 µs	大约 4µs	大约 0.8 µs
0	0	1	0	2^6 周期	大约 16 µs	大约 8µs	大约 1.6 µs
0	0	1	1	2^7 周期	大约 32 µs	大约 16µs	大约 3.2 µs
0	1	0	0	2^8 周期	大约 64 µs	大约 32µs	大约 6.4 µs
0	1	0	1	2^9 周期	大约 128 µs	大约 64µs	大约 12.8 µs
0	1	1	0	2^{10} 周期	大约 256 µs	大约 128µs	大约 25.6 µs
0	1	1	1	2^{11} 周期	大约 512 µs	大约 256µs	大约 51.2 µs
1	0	0	0	2^{12} 周期	大约 1.0 ms	大约 512 µs	大约 0.1 ms
1	0	0	1	2^{13} 周期	大约 2.0 ms	大约 1.0 ms	大约 0.2 ms
1	0	1	0	2^{14} 周期	大约 4.1ms	大约 2.0 ms	大约 0.4 ms
1	0	1	1	2^{15} 周期	大约 8.2ms	大约 4.1ms	大约 0.8 ms
1	1	0	0	2^{17} 周期	大约 33.0 ms	大约 16.4 ms	大约 3.3 ms
1	1	0	1	2^{19} 周期	大约 131 ms	大约 65.5 ms	大约 13.1 ms
1	1	1	0	2^{21} 周期	大约 524 ms	大约 262 ms	大约 52.4 ms
1	1	1	1	2^{23} 周期	大约 2.0 s	大约 1.0 s	大约 0.2 s

注意事项：

- *使能 SCM_CTL 寄存器的各振荡使能位 (SOSCE, MOSCE) 之前，设定各个振荡稳定等待时间。若各振荡器在等待震荡稳定时更改 MOWT 或 SOWT 位，不能保证各振荡稳定等待时间。*
- *软件复位不能初始化本寄存器。*

5.8 PLL 时钟稳定等待时间设置寄存器 (PSW_TMR)

PLL 时钟稳定等待时间设置寄存器 (PSW_TMR) 设置主 PLL 时钟的稳定等待时间。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	保留			PINC	保留	POWT		
属性		-		R/W	-	R/W		
初始值		-		0	-	000		

寄存器功能

[bit7:5] 保留：保留位

这些位的读取值为 0b000。写入时将这些位设置为 0b000。

[bit4] PINC: PLL 输入时钟选择位

位	描述
0	选择 CLKMO （主时钟振荡）[初始值]
1	选择 CLKHC（高速 CR 时钟）

注意事项：将此位设置为 "1" 有一定限制。

在注释列表 B 中查看当高速 CR 时钟作为主控时钟时的注释 1。

[bit3] 保留：保留位

此位读取值为 0. 写入时将此位设置为 0.

[bit2:0] POWT: 主 PLL 时钟稳定等待时间设置位

bit2	bit1	bit0	描述	计算示例 1 CLKPLL=20MHz	计算示例 2 CLKPLL=40MHz
0	0	0	2^9 周期 [初始值]	大约 25.6 µs	大约 12.8 µs
0	0	1	2^{10} 周期	大约 51.2 µs	大约 25.6 µs
0	1	0	2^{11} 周期	大约 102.4µs	大约 51.2 µs
0	1	1	2^{12} 周期	大约 204.8µs	大约 102.4µs
1	0	0	2^{13} 周期	大约 409.6µs	大约 204.8µs
1	0	1	2^{14} 周期	大约 819.2µs	大约 409.6µs
1	1	0	2^{15} 周期	大约 1638.4 µs	大约 819.2µs
1	1	1	2^{16} 周期	大约 3276.8µs	大约 1638.4 µs

注意事项：

- *使能 SCM_CTL 的 PLL 振荡使能位 (PLLE) 之前，设定各个振荡稳定等待时间。*
  若在 PLL 振荡器等待振荡稳定时更改 POWT 位，不能保证振荡稳定等待时间。
- *软件复位不能初始化本寄存器。*
- *主 PLL 时钟模式时如果 PINC 位为 1，无论 HCRE 位 (SCM_CTL:HCRE) 的值如何，均为高速 CR 时钟振荡使能状态。若 PINC 位为 0，高速 CR 时钟 (CLKHC) 的状态取决于 HCRE 位、FCSDE 位(CSV_CTL:FCSDE) 和 MCSVE 位 (CSV_CTL:MCSVE) 的设置。*
- *主 PLL 时钟模式时如果 PINC 位为 1，主时钟震荡状态取决于 MOSCE (SCM_CTL) 的设置。*
  若 PINC 位为 0，无论 MOSCE 位的值如何，都为主时钟震荡使能状态。

5.9 PLL 控制寄存器 1 (PLL_CTL1)

PLL 控制寄存器 1 (PLL_CTL1) 设置 PLL 的分频比。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	PLLK				PLLM			
属性	R/W				R/W			
初始值	0000				0000			

寄存器功能

[bit7:4] PLLK: PLL 输入时钟分频比设置位

bit 7:4	描述
0000	分频为 (PLLK 值 +1) (分频： 1 至 16） 示例： PLLK 值 (0000) +1 => 1/1 频率 [初始值]
0001	
.	
.	
1111	

[bit3:0] PLLM: PLL VCO 时钟分频比设置位

bit3:0	描述
0000	分频为 (PLLM 值 +1) (分频： 1 至 16） 示例： PLLM 值 (0000) +1 => 1/1 频率 [初始值]
0001	
.	
.	
1111	

注意事项：

- *使能 SCM_CTL 寄存器的 PLL 振荡使能位 (PLLE) 之前设置各分频比。*
- *软件复位不能初始化本寄存器。*

5.10 PLL 控制寄存器 2 (PLL_CTL2)

PLL 控制寄存器 2 (PLL_CTL2) 设置 PLL 的分频比。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	保留		PLLN					
属性	-		R/W					
初始值	-		000000					

寄存器功能

[bit7:6] 保留：保留位

这些位的读取值为 0b00。写入时将这些位设置为 0b00。

[bit5:0] PLLN: PLL 反馈分频比设置位

bit5:0	描述
000000	分频为 (PLLN 值 +1) (分频： 1 至 50） 示例： PLLN 值 (000000) +1 => 1/1 频率 [初始值]
000001	
.	
.	
110001	
110010	禁止设置
.	
111111	

注意事项：

- *使能 SCM_CTL 寄存器的 PLL 振荡使能位 (PLLE) 之前设置分频比。*
- *软件复位不能初始化本寄存器。*

5.11 调试断点看门狗计时器控制寄存器 (DBWDT_CTL)

调试断点看门狗计时器控制寄存器 (DBWDT_CTL) 设置调试模式工具断点的看门狗计时器计数操作。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	DPHWBE	保留	DPSWBE	保留				
属性	R/W	-	R/W	-				
初始值	0	-	0	-				

寄存器功能

[bit7] DPHWBE: HW-WDG 调试模式断点位

位	描述
0	HW-WDG 在工具断点停止计数 [初始值]
1	HW-WDG 在工具断点继续计数

[bit6] 保留：保留位

此位读取值为 0. 写入时将此位设置为 0.

[bit5] DPSWBE: SW-WDG 调试模式断点位

位	描述
0	SW-WDG 在工具断点停止计数 [初始值]
1	SW-WDG 在工具断点继续计数

[bit4:0] 保留：保留位

这些位的读取值为 0b00000。写入时将这些位设置为 0b00000。

注意事项：

- *软件复位不能初始化本寄存器。*

5.12 中断使能寄存器(INT_ENR)

中断使能寄存器 (INT_ENR) 使能/禁用中断。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	保留		FCSE	保留		PCSE	SCSE	MCSE
属性	-		R/W	-		R/W	R/W	R/W
初始值	-		0	-		0	0	0

寄存器功能

[bit7:6] 保留：保留位

这些位的读取值为 0b00。写入时将这些位设置为 0b00。

[bit5] FCSE: 异常频率检测中断使能位

位	描述
0	禁用 FCS 中断
1	使能 FCS 中断

[bit4:3] 保留：保留位

这些位的读取值为 0b00。写入时将这些位设置为 0b00。

[bit2] PCSE: PLL 振荡稳定等待完成中断使能位

位	描述
0	禁用 PLL 振荡稳定等待完成中断
1	使能 PLL 振荡稳定等待完成中断

[bit1] SCSE: 副时钟振荡稳定等待完成中断使能位

位	描述
0	禁用副时钟振荡稳定等待完成中断
1	使能副时钟振荡稳定等待完成中断

[bit0] MCSE: 主时钟振荡稳定等待完成中断使能位

位	描述
0	禁用主时钟振荡稳定等待完成中断
1	使能主时钟振荡稳定等待完成中断

注意事项：

- *有关异常频率检测，参见时钟监视器一章。*

5.13 中断状态寄存器(INT_STR)

中断状态寄存器 (INT_STR) 指示中断的状态。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	保留		FCSI	保留		PCSI	SCSI	MCSI
属性	-		R	-		R	R	R
初始值	-		0	-		0	0	0

寄存器功能

[bit7:6] 保留：保留位

这些位的读取值为 0b00。写入时将这些位设置为 0b00。

[bit5] FCSI: 异常频率检测中断状态位

位	描述
0	FCS 中断未生效
1	FCS 中断已生效

[bit4:3] 保留：保留位

这些位的读取值为 0b00。写入时将这些位设置为 0b00。

[bit2] PCSI: PLL 振荡稳定等待完成中断状态位

位	描述
0	PLL 振荡稳定等待完成中断未生效
1	PLL 振荡稳定等待完成中断已生效

[bit1] SCSI: 副时钟振荡稳定等待完成中断状态位

位	描述
0	副时钟振荡稳定等待完成中断未生效
1	副时钟振荡稳定等待完成中断已生效

[bit0] MCSI: 主时钟振荡稳定等待完成中断状态位

位	描述
0	主时钟振荡稳定等待完成中断未生效
1	主时钟振荡稳定等待完成中断已生效

5.14 中断清除寄存器(INT_CLR)

中断清除寄存器 (INT_CLR) 清除中断因素。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	保留		FCSC	保留		PCSC	SCSC	MCSC
属性	-		W	-		W	W	W
初始值	-		0	-		0	0	0

寄存器功能

[bit7:6] 保留：保留位

这些位的读取值为 0b00。写入时将这些位设置为 0b00。

[bit5] FCSC: 异常频率检测中断因素清除位

位	描述
写入 0 时	写入值不影响 FCS 中断因素
写入 1 时	清除 FCS 中断因素
读取时	读取值为固定值 "0"。

[bit4:3] 保留：保留位

这些位的读取值为 0b00。写入时将这些位设置为 0b00。

[bit2] PCSC: PLL 振荡稳定等待完成中断因素清除位

位	描述
写入 0 时	写入值不影响 PLL 振荡稳定等待完成中断因素。
写入 1 时	清除 PLL 振荡稳定等待完成中断因素。
读取时	读取值为固定值 "0"。

[bit1] SCSC: 副时钟振荡稳定等待完成中断因素清除位

位	描述
写入 0 时	写入值不影响副时钟振荡稳定等待完成中断因素。
写入 1 时	清除副时钟振荡稳定等待完成中断因素。
读取时	读取值为固定值 "0"。

[bit0] MCSC: 主时钟振荡稳定等待完成中断因素清除位

位	描述
写入 0 时	写入值不影响主时钟振荡稳定等待完成中断因素。
写入 1 时	清除主时钟振荡稳定等待完成中断因素。
读取时	读取值为固定值 "0"。

注意事项：

- *清除此寄存器后，也清除了 INT_STR 寄存器中的各中断状态位 (FCSI, PCSI, SCSI, MCSI)*

6. 使用注意事项

本节说明使用时钟生成单元的注意事项。

■ 主时钟和副时钟振荡器的振荡稳定等待时间

由于主时钟/副时钟振荡器的稳定等待时间取决于振荡器类型（晶振型、陶瓷型等），必须选择适合于各类型振荡器的振荡稳定等待时间。

■ 更改稳定 PLL 振荡下的分频。

在 PLL 振荡稳定后更改 PLL 分频比时，停止 PLL 振荡，更改分频比并重新使能 PLL 振荡。

■ 外围设备不受时钟生成单元的时钟控制

以下外围设备运行不受时钟生成单元的时钟控制。

有关各操作时钟的信息，参见以章节：

- USB 时钟生成单元： 参见 "通信宏部分"的 “USB 时钟生成”一章。
- 时钟监视器 : 参见时钟监视器一章
- 看门狗计时器 : 参见计时器部分的看门狗计时器一章。
- 计时计数器 : 参见计时器部分的计时计数器一章。
- 实时时钟 : 参见计时器部分的实时时钟一章。
- CAN 预分频器 : 参见通信宏部分的 CAN 预分频器一章。

■ 设定振荡稳定等待时间

通过相关振荡稳定等待时间设置寄存器，设置主时钟、副时钟和 PLL 振荡器的振荡稳定等待时间，然后使能各个振荡器。

等待振荡稳定时不得更改振荡稳定等待时间。

■ 在使用主 PLL 时钟期间检查主时钟振荡。

禁止在使用 PLL 振荡时停止主时钟振荡。

■ 切换时钟模式

可通过更改 SCM_CTL 寄存器的 RCS[2:0] 位来切换时钟模式。

切换时钟模式的步骤如下：

1. 设置各个振荡器的振荡稳定等待时间。
2. 将期望时钟的振荡使能位 (SCM_CTL:xxxE) 设置为 "1"。
3. 检查期望时钟的振荡稳定位 (SCM_CTL:xxxRDY) 是否为 "1"。
4. 切换 SCM_CTL:RCS[2:0]。
5. 等待直至 SCM_STR:RCM[2:0] = SCM_CTL:RCS[2:0]。

■ 时钟模式切换和振荡稳定位之间的关联。

振荡稳定位 (SCM_STR:xxxRDY) 变为 1 时的时间随以下时钟模式切换而变化：

- 从高速 CR 运行、主运行或 PLL 运行切换至另一时钟模式时：

  将 SCM_CTL:xxxE 设为 1 可启动振荡稳定等待时间。在振荡稳定等待时间结束后，可检查 SCM_STR:xxxRDY 是否为 1。

- 从低速 CR 运行或副运行切换至高速 CR 运行、主运行或 PLL 运行时：

  即使 SCM_CTL:MOSCE（或 PLLE）设为 1，主时钟振荡也不会启动。启动主时钟（或高速 CR 或 PLL）振荡稳定等待时间时，必须在 SCM_CTL:MOSCE（或 PLLE）设置为 1 后切换 SCM_CTL:RCS [2:0]。振荡稳定等待时间结束后，可检查 SCM_STR:xxxRDY 是否为 1。

- 如果待机模式被中断释放，从机将 SCM_CTL 寄存器中 RCS[2:0] 位所指示的在时钟模式下重新启动。

- 若发生软件复位之外的其他复位，高速 CR 时钟 (CLKHC) 设置为主控时钟。将高速 CR 时钟模式设置为时钟模式。
- 若执行软件复位之外的其他复位，主时钟和副时钟振荡器以及 PLL 振荡停止。若要在复位后再次使用这些振荡器，则使用 SCM_CTL 寄存器来使能这些振荡器。
- 有关各时钟模式和振荡器启动/停止之间的关联，参见低功耗模式一章。
- 关闭芯片侧的电源，只运行 VBAT 域时，需设置 WTOSCCNT.SOSCNTL=0，然后再关闭芯片侧的电源。

第 2-2 章：外设时钟门控

本章说明外设时钟门控的功能。

1. 外设时钟门控概述
2. 外设时钟门控配置
3. 外设时钟门控
4. 外设时钟门控功能寄存器
5. 外设时钟门控功能使用注意事项

代码：9BFPCG-FM0-C03.0

1. 外设时钟门控概述

本节概述单独停止外设功能操作时钟的外设时钟门控。本系统可通过对无需使用的外设进行时钟门控来降低整个系统的电流消耗。

外设时钟门控

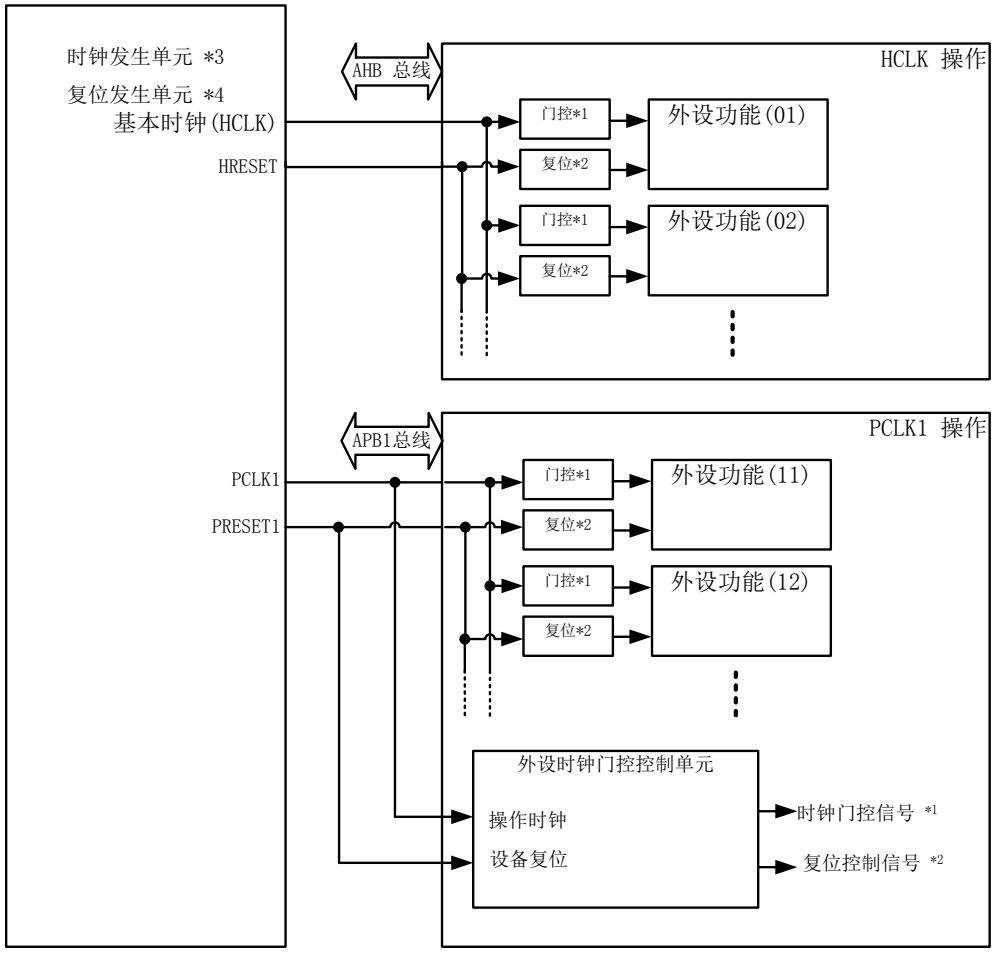
- 对系统操作时未使用的外设功能的操作时钟单独进行门控。
- 关于目标时钟及外设时钟门控单元，参见“门控单元及其外设时钟初始状态”一章
- 时钟门控时或提供时钟前，可复位外设功能的内部状态。

通过设置连接到 APB1 总线的寄存器执行上述外设时钟门控和复位控制。

时钟与复位生成单元间的连接

Figure 1-1 所示为外设时钟门控与时钟生成单元或复位生成单元之间的连接。外设时钟门控存在于外设功能与时钟生成单元或复位生成单元以及外设功能单元的门控时钟与控制复位之间。复位控制单元停止提供内部总线时钟时，优先设置时钟控制单元，并门控向外设功能提供操作时钟。使用外设时钟门控时，确保设置使能时钟生成单元中的 APB1 总线时钟 (PCLK1) ，以控制其他时钟。

Figure 1-1 外设时钟门控相关的时钟/复位连接



*1: 外设功能单元中的时钟门控（内部总线时钟门控）
*2: 外设功能单元的强制复位和总线复位(参见“复位”一章)
*3: 详见“时钟”一章
*4: 详见“复位”一章

门控单元及其外设时钟的初始状态

门控单元及其外设时钟门控的初始状态，参见 Table 1-1。

Table 1-1 控制单元及其外设时钟门控的初始状态

外设功能	停止时钟的结构单元	初始状态	备注
多功能串口	一通道	时钟提供	
基本计时器	四通道	时钟提供	通过每四个通道（Ch.0-Ch.3、Ch.4-Ch.7、Ch.8-Ch.11 以及 Ch.12-Ch.15）控制时钟门控。
多功能计时器	一单元	时钟提供	-
PPG	八通道	时钟提供	通过每八个通道（Ch.0-Ch.7、Ch.8-Ch.15、Ch.16-Ch.23 以及 Ch.24-Ch.31）控制时钟门控。
四位计数器	一单元	时钟提供	-
DMAC	一单元	时钟提供	-
CAN 控制器	一通道	时钟提供	-
A/D 转换器	一单元	时钟提供	-
IO 口 (GPIO/快速 GPIO)	所有端口	时钟提供	时钟门控限制条件，参见 5. 外设时钟门控一节中的“使用注意事项”。
DSTC	一单元	时钟提供	-
智能卡接口（IC 卡）	一通道	时钟提供	-
MFSI2S	一通道	时钟提供	-
LCD 控制器	一单元	时钟提供	-
HDMI-CEC/遥控接收器	一单元	时钟提供	-
可编程 CRC	一单元	时钟提供	-
USB（从机/主机）	一通道	时钟停止	-

注意事项：

- *有关所安装外设功能的类型及数量，参见所用产品的《数据手册》。*
- *PPG 的时钟控制与多功能计时器共用设置位。详细信息参见 4.3 外设时钟控制寄存器 1 (CKEN1)。*
- *单独与“5. DSTC 中的 DSTC 寄存器”执行 DSTC 单元的时钟控制。*

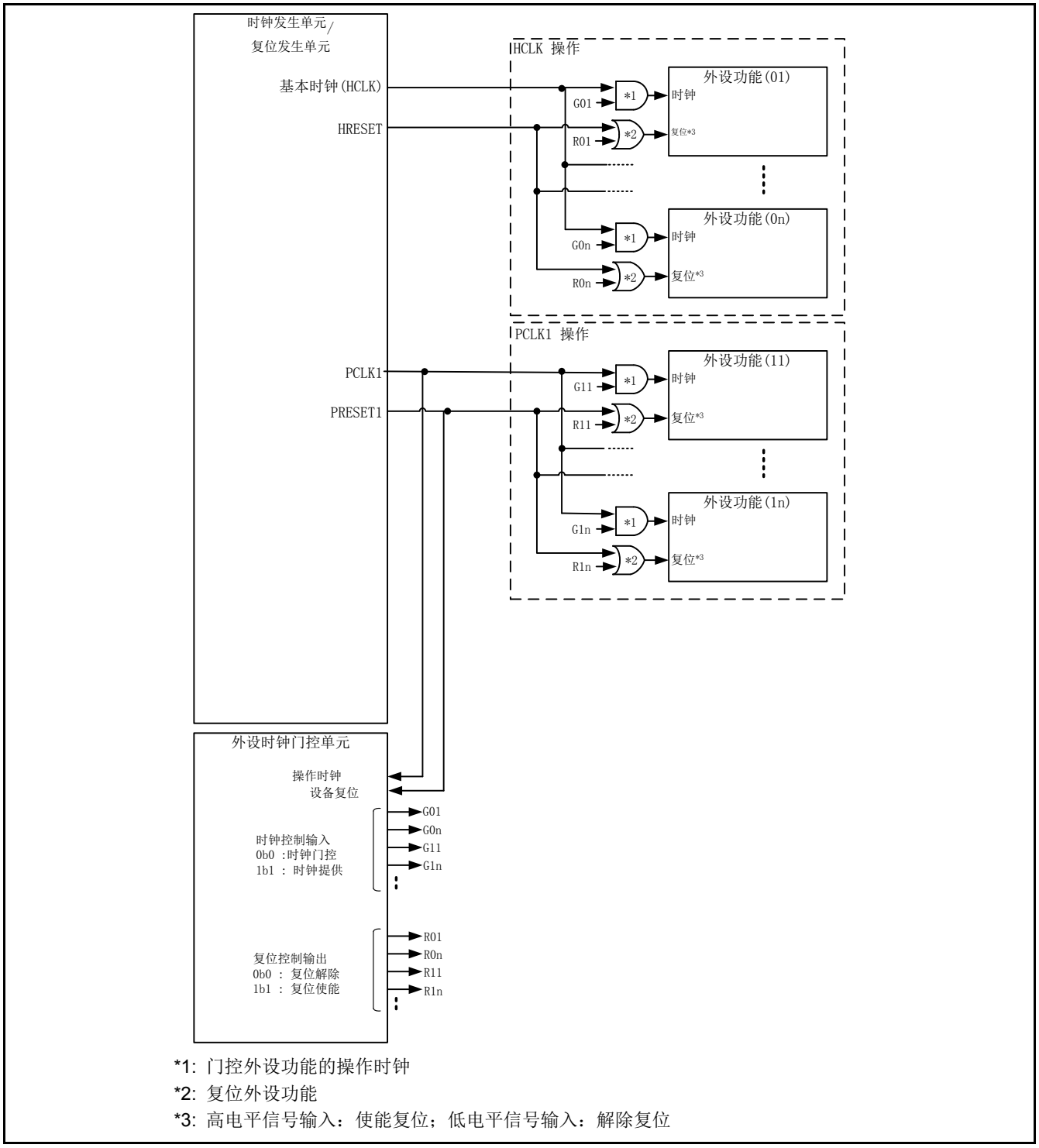
2. 外设时钟门控配置

本章说明外设时钟门控的配置。

方框图

Figure 2-1 所示为外设时钟门控的系统配置。

Figure 2-1 外设时钟门控方框图



*1: 门控外设功能的操作时钟
*2: 复位外设功能
*3: 高电平信号输入：使能复位；低电平信号输入：解除复位

方框图说明

■ 外设时钟门控控制单元

通过改变经由 APB1 总线的寄存器设置值，执行各外设功能的时钟控制或复位控制。确保通过设置时钟控制单元的 APB1 预分频寄存器 (APBC1_PSR) 内 APB1 时钟使能位 (APBC1EN) 为输出使能并允许 PCLK1 输出，而重写本寄存器。

目标功能的位字段设置为 0 时，各外设功能时钟停止。位字段设置为 1 时，提供时钟。不同外设功能的寄存器初始值各不相同。有关详细信息，参见 Table 1-1。

目标功能的位字段设置为 1 时，各外设功能可复位。位字段设置为 "0" 时，释放复位。释放复位时，各寄存器的初始值总为 0。

■ 外设时钟门控逻辑

根据外设时钟门控单元的时钟门控信号由各特定外设功能提供或门控内部总线时钟 (HCLK、PCLK1)。

■ 外设复位控制逻辑

各外设功能分别根据外设时钟门控控制单元发出的复位控制信号控制复位。复位控制单元与外设时钟控制单元相同，但没有单独用于 I/O 口的这种复位控制专用位字段。

3. 外设时钟门控

本章说明外设时钟门控。

通过总线复位 (PRESET1)* 使外设时钟门控的寄存器复位为初始状态。因总线复位 (PRESET1) 由所有复位因素生成，总线复位后必须立即执行必要外设功能的时钟控制。

*: 有关总线复位 (PRESET1) 的生成条件，参见复位一章。

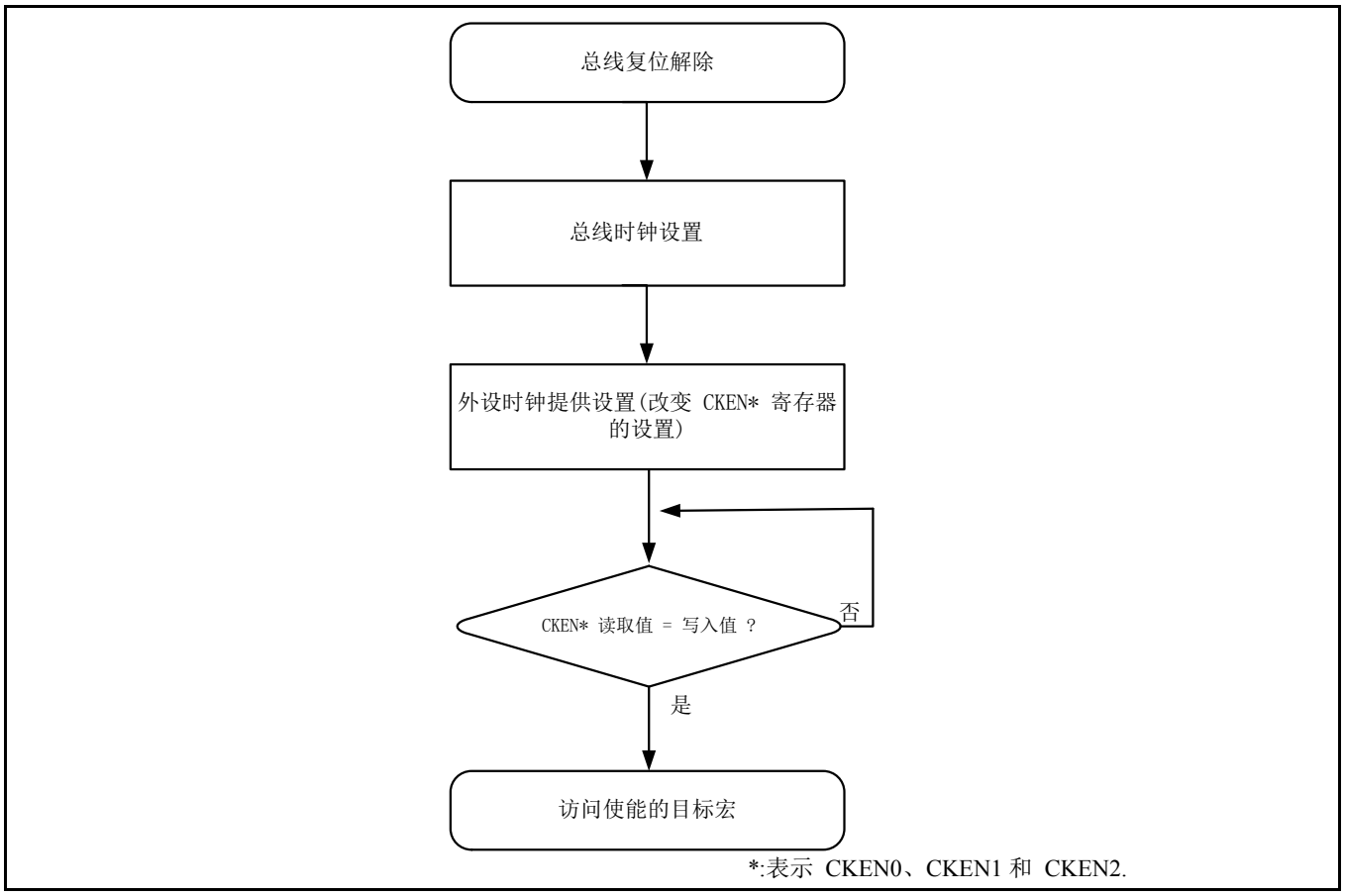
3.1 外设时钟控制步骤

本节说明外设时钟提供和停止的控制步骤。

时钟提供步骤

总线复位释放后立即将总线时钟和外设时钟的设定值复位为初始值。因此，按照 Figure 3-1 所述步骤为初始状态停止的外设功能时钟设置时钟提供。

Figure 3-1 时钟提供步骤



1. 总线时钟设置

   使用时钟生成部分的寄存器执行各总线的设置。

   有关设置的详细信息，参见时钟一章。

2. 外设时钟的提供设置

   变更外设功能相对应的位设置，为初始时钟门控状态下时钟控制的外设时钟控制寄存器（CKEN0、CKEN1 及 CKEN2）提供时钟。

3. 外设时钟控制寄存器的设定值确认

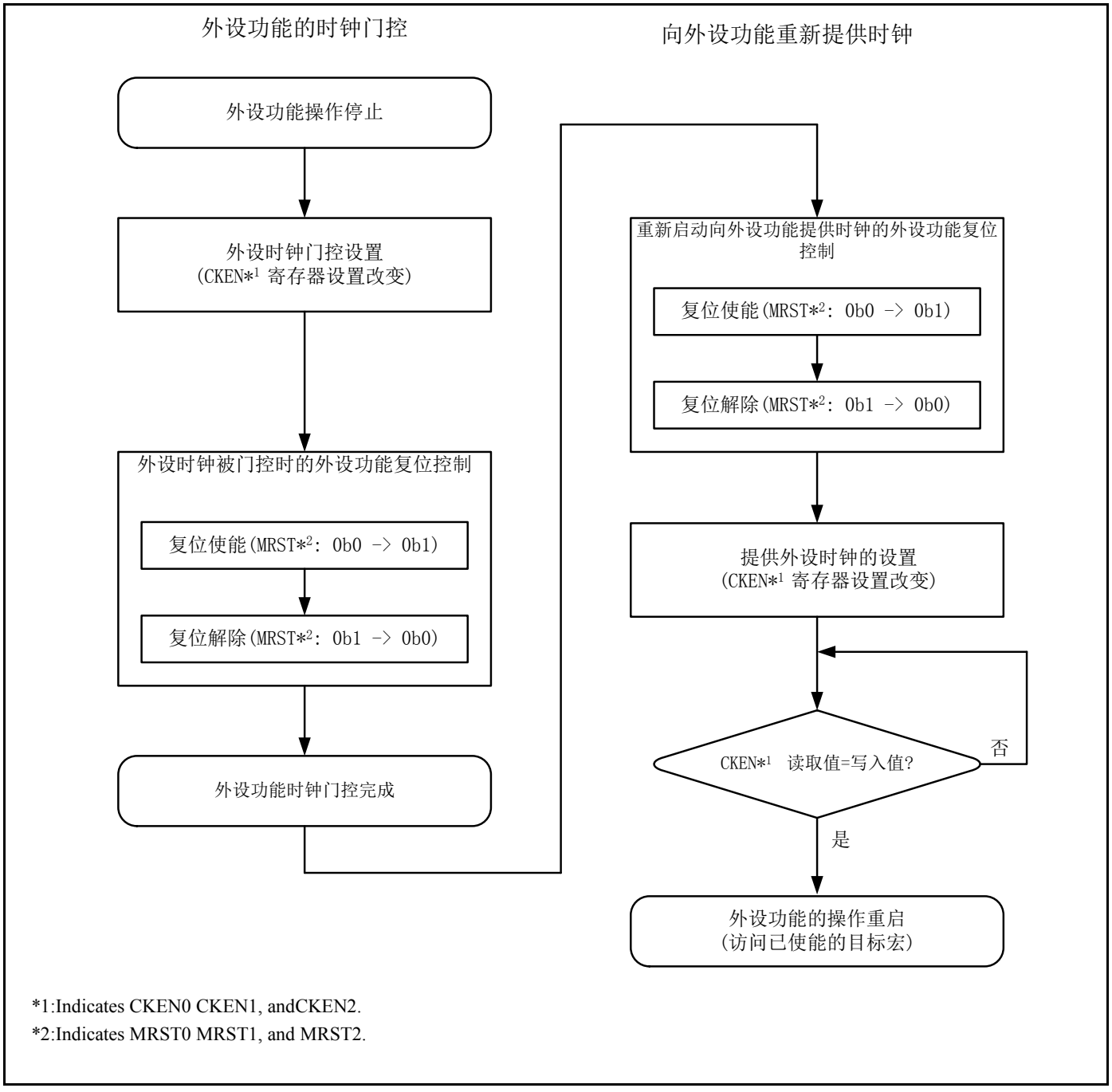
外设时钟寄存器（CKEN0、CKEN1 及 CKEN2）将值更新为向变更设置的外设功能提供时钟启动步骤时所写入的值。

由于时钟门控时访问外设功能无效，因此要确保在上述第 2 项设置变更后启动外设功能的访问，读取此寄存器，然后确认与写入值一致。

时钟门控和再提供步骤

Figure 3-2 说明了外设功能时钟门控步骤以及向外设功能再提供时钟的步骤。

Figure 3-2 外设功能时钟门控及向外设功能再提供时钟的步骤



*1:Indicates CKEN0 CKEN1, andCKEN2.
*2:Indicates MRST0 MRST1, and MRST2.

■ 外设功能时钟门控

1. 外设时钟门控设置

对于外设时钟控制寄存器（CKEN0、CKEN1 及 CKEN2），变更时钟提供将停止至 0 的外设功能对应位。

根据时钟门控指示进行外设功能时钟门控后，外设时钟控制寄存器（CKEN0、CKEN1 及 CKEN2）将寄存器值更新为写入值。

2. 外设时钟被门控的外设功能的复位控制

对于时钟被门控的外设功能，复位其内部状态时按以下步骤执行各外设功能的复位控制。

复位使能：

外设功能复位控制寄存器（MRST0、MRST1 及 MRST2）的目标位写入 1。

释放复位：

外设功能复位控制寄存器（MRST0、MRST1 及 MRST2）的目标位写入 0 。

■ 外设功能时钟再提供

1. 重启向外设功能提供时钟的外设功能复位控制

门控外设时钟的外设功能重启操作前，使用外设功能复位控制寄存器（MRST0、MRST1 及 MRST2）执行各外设功能的复位控制。步骤与上述外设时钟门控后立即复位控制的步骤相同。

2. 外设时钟提供设置

对于外设时钟控制寄存器（CKEN0、CKEN1 及 CKEN2），变更再提供时钟的外设功能对应位的设置。

此时，勿设置未提供外设功能的位以及总线时钟已门控至非初始值的位。其原因是读取值无法与未能从下述第 3 项中寄存器设定值确认的处理循环获得的写入值相吻合。

3. 外设时钟控制寄存器设定值的确认

在时钟设置变化反映到改变设置的外设功能上这一步骤时，外设时钟控制寄存器（CKEN0、CKEN1 及 CKEN2）将寄存器值更新为写入值。

由于时钟门控时外设功能访问无效，确保执行上述第 2 项的设置变更后启动外设功能访问，读取寄存器值，然后确认与写入值一致。

4. 外设时钟门控功能寄存器

本节说明外设时钟门控功能的各寄存器功能。

Table 4-1 所示为外设时钟门控功能的寄存器列表。

Table 4-1 外设时钟门控功能寄存器

寄存器名称缩写	寄存器名称	参考章节
CKEN0	外设功能时钟控制寄存器 0	4.1
MRST0	外设功能复位控制寄存器 0	4.2
CKEN1	外设功能时钟控制寄存器 1	4.3
MRST1	外设功能复位控制寄存器 1	4.4
CKEN2	外设功能时钟控制寄存器 2	4.5
MRST2	外设功能复位控制寄存器 2	4.6

4.1 外设功能时钟控制寄存器 0 (CKEN0)

本节说明外设功能复位时钟寄存器 0 (CKEN0)。

位	31	30	29	28	27	26	25	24
字段	保留			GIOCK	保留		DSTCCK	DMACK
属性		-		R/W		-	R/W	R/W
初始值		-		1		-	1	1

位	23	22	21	20	19	18	17	16
字段	保留				ADCCK[3:0]			
属性		-			R/W			
初始值		-			1111			

位	15	14	13	12	11	10	9	8
字段	MFSCK[15:8]							
属性	R/W							
初始值	0xFF							

位	7	6	5	4	3	2	1	0
字段	MFSCK[7:0]							
属性	R/W							
初始值	0xFF							

[bit31:29] 保留：保留位

将 0 写入这些位。

[bit28] GIOCK： GPIO/快速 GPIO 功能的软件时钟控制

此位控制针对 I/O 口功能的操作时钟的提供及门控。此位统一控制提供给 I/O 口功能的所有操作时钟。

此位设置为 1 时，向 I/O 口功能模块提供总线时钟，并可使用 I/O 口功能。

此位设置为 0 时，向 I/O 口功能模块输入的总线时钟受控。注意，总线时钟受到门控时，输入电平的读取以及输出功率电平的设置变更被禁用。详细信息参见“5. 外设时钟门控功能使用注意事项”。

位	描述
0	向 I/O 口功能模块输入总线时钟受控。
1	向 I/O 口功能模块提供总线时钟。（初始值） 确认设置为"1"以便使用 I/O 口功能。

[bit27:26] 保留：保留位

将 0 写入此位。

[bit25] DSTCCK：DSTC 功能的软件时钟控制

此位控制针对 DSTC 功能的操作时钟的提供及门控。此位设置为 1 时，向 DSTC 块提供总线时钟，并可使用 DSTC 功能。

此位设置为 0 时，向 DSTC 模块输入的总线时钟受控。总线时钟输入受控时，不能使用 DSTC 功能。

位	描述
0	向 DSTC 功能模块输入总线时钟受控。
1	向 DSTC 功能模块提供总线时钟。（初始值） 确认设置为"1"以便使用 DSTC 功能。

注意事项：TYPE 3-M0+ 产品中此位值被忽略。 DSTC 中的待机控制寄存器可用于 DSTC 的时钟停止。

[bit24] DMACK：DMAC 操作时钟的提供及门控设置

此位控制针对 DMAC 功能的操作时钟的提供及门控。此位设置为 1 时，向 DMAC 块提供总线时钟，并可使用 DMAC 功能。

此位设置为 0 时，门控输入至 DMAC 模块的总线时钟。门控总线时钟输入时，不能使用 DMAC 功能。

位	描述
0	门控 DMAC 进行总线时钟输入。
1	向 DMAC 提供总线时钟。（初始值）

[bit23:20] 保留：保留位

将 0 写入这些位。

[bit19:16] ADCCK[3:0]：对 AD 转换器提供及门控操作时钟的设置

这些位控制针对 AD 转换器的操作时钟的提供及门控。以下为各位与 AD 转换器单元的对应关系：

bit16 - ADCCK0：AD 转换器单元 0

bit17 - ADCCK1：AD 转换器单元 1

bit18 - ADCCK2：AD 转换器单元 2

bit19 - ADCCK3：AD 转换器单元 3

相关位设置为 1 时，向对应的 AD 转换器单元提供总线时钟，以使能 AD 转换器功能。对于未安装对应 AD 转换器的产品，不得变更其相关位的初始值。

相关位设置为 0 时，门控对应 AD 转换器的总线时钟输入。门控总线时钟输入时，不能使用相关的 AD 转换器。

位	描述
0	门控相关位对应的 A/D 转换器单元的总线时钟输入。
1	向相关位对应的 A/D 转换器单元提供总线时钟。（初始值）

[bit15:0] MFSCK[15:0]：对多功能串口提供及门控操作时钟的设置

这些位控制针对多功能串口的操作时钟的提供及门控。以下为各位与通道的对应关系：

bit0 - MFSCK0：多功能串口通道 0

bit1 - MFSCK1：多功能串口通道 1

bit2 - MFSCK2：多功能串口通道 2

bit3 - MFSCK3：多功能串口通道 3

bit4 - MFSCK4：多功能串口通道 4

bit5 - MFSCK5：多功能串口通道 5

bit6 - MFSCK6：多功能串口通道 6

bit7 - MFSCK7：多功能串口通道 7

bit8 - MFSCK8：多功能串口通道 8

bit9 - MFSCK9：多功能串口通道 9

bit10 - MFSCK10：多功能串口通道 10

bit11 - MFSCK11：多功能串口通道 11

bit12 - MFSCK12：多功能串口通道 12

bit13 - MFSCK13：多功能串口通道 13

bit14 - MFSCK14：多功能串口通道 14

bit15 - MFSCK15：多功能串口通道 15

相关位设置为 1 时，向对应多功能串口的通道提供总线时钟，以使能多功能串口的功能。对于未安装相关多功能串口通道的产品，不得变更其相关位的初始值。

相关位设置为 0 时，门控对应多功能串口的通道的总线时钟输入。门控总线时钟输入时，不能使用对应通道的多功能串口功能。

位	描述
0	门控相关位对应多功能串口通道的总线时钟输入。
1	向相关位对应的多功能串口通道提供总线时钟。（初始值）

4.2 外设复位控制寄存器 0 (MRST0)

本节说明外设复位控制寄存器 0 (MRST0)。

位	31	30	29	28	27	26	25	24
字段	保留						DSTCRST	DMARST
属性	-						R/W	R/W
初始值	-						0	0

位	23	22	21	20	19	18	17	16
字段	保留				ADCRST[3:0]			
属性	-				R/W			
初始值	-				0000			

位	15	14	13	12	11	10	9	8
字段	MFSRST [15:8]							
属性	R/W							
初始值	0x00							

位	7	6	5	4	3	2	1	0
字段	MFSRST [7:0]							
属性	R/W							
初始值	0x00							

[bit31:26] 保留：保留位

将 0 写入这些位。

[bit25] DSTCRST：DSTC 的复位控制

此位控制 DSTC 单元的复位。如此位设置为 1，DSTC 变为复位状态，DSTC 的 DMA 传输操作停止，且所有寄存器设置初始化。释放复位状态需再次将此位设置为 0。

位	描述
0	释放 DSTC 复位。（初始值）
1	向 DSTC 发出复位信号。

注意事项：TYPE 3-M0+ 产品中此位值被忽略。

[bit24] DMARST：DMAC 的复位控制

此位控制 DMAC 单元的复位。如此位设置为 1，DMAC 变为复位状态，DMA 传输操作停止，且所有寄存器设置初始化。释放复位状态需再次将此位设置为 0。

位	描述
0	解除 DMAC 复位。（初始值）
1	向 DMAC 发出复位信号。

[bit23:20] 保留：保留位
将 0 写入这些位。

[bit19:16] ADCRST[3:0]：AD 转换器的复位控制
这些位控制 AD 转换器各单元的复位。以下为各位与 AD 转换器单元的对应关系：

bit16 - ADCRST 0：AD 转换器单元 0
bit17 - ADCRST 1：AD 转换器单元 1
bit18 - ADCRST 2：AD 转换器单元 2
bit19 - ADCRST 3：AD 转换器单元 3

如相关位设置为 1，对应 AD 转换器单元变为复位状态，数字转换模拟操作停止，且寄存器设置初始化。对于未安装相关 AD 转换器单元的产品，不得变更其相关位的初始状态。释放复位状态需再次将此位设置为 0。

位	描述
0	释放相关位对应的 AD 转换器单元的复位。（初始值）
1	向相关位对应的 AD 转换器单元发出复位信号。

[bit15:0] MFSRST[15:0]：多功能串口的复位控制
这些位控制多功能串口各通道的复位。以下为各位与通道的对应关系：

bit0 - MFSRST0：多功能串口通道 0
bit1 - MFSRST1：多功能串口通道 1
bit2 - MFSRST2：多功能串口通道 2
bit3 - MFSRST3：多功能串口通道 3
bit4 - MFSRST4：多功能串口通道 4
bit5 - MFSRST5：多功能串口通道 5
bit6 - MFSRST6：多功能串口通道 6
bit7 - MFSRST7：多功能串口通道 7
bit8 - MFSRST8：多功能串口通道 8
bit9 - MFSRST9：多功能串口通道 9
bit10 - MFSRST10：多功能串口通道 10
bit11 - MFSRST11：多功能串口通道 11
bit12 - MFSRST12：多功能串口通道 12
bit13 - MFSRST13：多功能串口通道 13
bit14 - MFSRST14：多功能串口通道 14
bit15 - MFSRST15：多功能串口通道 15

如相关位设置为 1，对应多功能串口的通道变为复位状态，其串行通讯停止，且寄存器设置初始化。对于未安装相关多功能串口通道的产品，禁止将其相关位设置为 1。释放上述复位需再次将此位设置为 0。

位	描述
0	释放相关位对应多功能串口通道的复位。（初始值）
1	向相关位对应的多功能串口通道发出复位信号。

4.3 外设时钟控制寄存器 1 (CKEN1)

本节说明外设时钟控制寄存器 1 (CKEN1)。

位	31	30	29	28	27	26	25	24
字段	保留							
属性	-							
初始值	-							

位	23	22	21	20	19	18	17	16
字段	保留				QDUCK[3:0]			
属性	-				R/W			
初始值	-				1111			

位	15	14	13	12	11	10	9	8
字段	保留				MFTCK[3:0]			
属性	-				R/W			
初始值	-				1111			

位	7	6	5	4	3	2	1	0
字段	保留				BTMCK[3:0]			
属性	-				R/W			
初始值	-				1111			

[bit31:20] 保留：保留位

将 0 写入这些位。

[bit19:16] QDUCK[3:0]：对四位计数器提供和门控操作时钟的设置

这些位控制针对四位计数器的操作时钟的提供及门控。以下为各位与四位计数器的对应关系：

bit16 - QDUCK0：四位计数器单元 0

bit17 - QDUCK1：四位计数器单元 1

bit18 - QDUCK2：四位计数器单元 2

bit19 - QDUCK3：四位计数器单元 3

相关位设置为 1 时，向对应的四位计数器单元提供总线时钟，以使用四位计数器功能。对于未安装相关四位计数器的产品，不得变更其相关位的初始值。

相关位设置为 0 时，停止向对应四位计数器输入总线时钟。门控总线时钟输入时，不能使用相关单元的四位计数器。

位	描述
0	门控向对应的四位计数器输入总线时钟。
1	向相关位对应的四位计数器提供总线时钟。（初始值）

[bit15:12] 保留：保留位
将 0 写入这些位。

[bit11:8] MFTCK[3:0]：对多功能计时器和 PPG 提供和门控操作时钟的设置
这些位控制针对多功能计时器和 PPG 的操作时钟的提供及门控。以下为各位与多功能计时器单元及 PPG 通道的对应关系：

bit8 - MFTCK0：多功能计时器单元 0 - PPG 通道 0、2、4、6

bit9 - MFTCK1：多功能计时器单元 1 - PPG 通道 8、10、12、14

bit10 - MFTCK2：多功能计时器单元 2 - PPG 通道 16、18、20、22

bit11 - MFTCK3：多功能计时器单元 3 - PPG 通道 24、26、28、30

相关位设置为 1 时，向对应的多功能计时器单元及 PPG 通道提供总线时钟，以使用多功能计时器及 PPG 功能。对于未安装相关多功能计时器单元及 PPG 通道的产品，禁止变更其相关位的初始值。
相关位设置为 0 时，门控向对应的多功能计时器单元及 PPG 通道输入总线时钟。门控总线时钟时，不能使用相关的多功能计时器及 PPG 功能。

位	描述
0	门控向相关位对应的多功能计时器单元和 PPG 通道输入总线时钟。
1	向相关位对应的多功能计时器单元和 PPG 通道提供总线时钟。（初始值）

[bit7:4] 保留：保留位
将 0 写入这些位。

[bit3:0] BTMCK[3:0]：对基本计时器门控提供和门控制操作时钟的设置
这些位控制针对基本计时器的操作时钟的提供及门控以下为各位与基本计时器通道的对应关系：

bit0 - BTMCK0：基本计时器通道 0、1、2、3

bit1 - BTMCK1：基本计时器通道 4、5、6、7

bit2 - BTMCK2：基本计时器通道 8、9、10、11

bit3 - BTMCK3：基本计时器通道 12、13、14、15

相关位设置为 1 时，向对应基本计时器通道提供总线时钟，以使用基本计时器。不得变更其相关位的初始值。

相关位设置为 0 时，门控向对应基本计时器通道输入总线时钟。门控总线时钟输入时，不能使用对应通道的基本计时器功能。

位	描述
0	门控向相关位对应的基本计时器通道输入总线时钟。
1	向相关位对应的基本计时器通道提供总线时钟。（初始值）

4.4 外设功能复位控制寄存器 1 (MRST1)

本节说明外设功能复位控制寄存器 1 (MRST1)。

位	31	30	29	28	27	26	25	24
字段	保留							
Attribute	-							
Initial value	-							

位	23	22	21	20	19	18	17	16
字段	保留				QDURST[3:0]			
属性	-				R/W			
初始值	-				0000			

位	15	14	13	12	11	10	9	8
字段	保留				MFTRST[3:0]			
属性	-				R/W			
初始值	-				0000			

位	7	6	5	4	3	2	1	0
字段	保留				BTMRST[3:0]			
属性	-				R/W			
初始值	-				0000			

[bit31:20] 保留：保留位
将 0 写入这些位。

[bit19:16] QDURST[3:0]：四位计数器的复位控制
这些位控制四位计数器各单元的复位。以下为各位与四位计数器的对应关系单元：

bit16 - QDURST 0：四位计数器单元 0
bit17 - QDURST 1：四位计数器单元 1
bit18 - QDURST 2：四位计数器单元 2
bit19 - QDURST 3：四位计数器单元 3

如相关位设置为 1，对应的四位计数器单元变为复位状态，四位计数器停止运行，且寄存器设置初始化。对于未安装相关四位计数器的产品，不得变更其相关位的初始状态。释放复位状态需再次将此位设置为 0。

位	描述
0	释放相关位对应的四位计数器的复位。（初始值）
1	向相关位对应的四位计数器单元发出复位信号。

[bit15:12] 保留：保留位
将 0 写入这些位。

[bit11:8] MFTRST[3:0]：多功能计时器的控制及 PPG 复位控制
这些位控制多功能计时器各单元的复位及每四个通道的 PPG 复位。以下为各位以及四位计数器单元和 PPG 通道的对应关系：

bit8 - MFTRST0：多功能计时器单元 0 - PPG 通道 0、2、4、6

bit9 - MFTRST1：多功能计时器单元 1 - PPG 通道 8、10、12、14

bit10 - MFTRST2：多功能计时器单元 2 - PPG 通道 16、18、20、22

bit11 - MFTRST3：多功能计时器单元 3 - PPG 通道 24、26、28、30

如相关位设置为 1，对应多功能计时器单元及 PPG 通道变为复位状态，多功能计时器运行停止，且寄存器设置初始化。对于未安装相关多功能计时器单元及 PPG 通道的产品，不得变更其相关位的初始值。释放复位状态需再次将此位设置为 0。

位	描述
0	释放相关位对应的多功能计时器和 PPG 通道的复位。（初始值）
1	向相关位对应的多功能计时器和 PPG 通道发出复位信号。

[bit7:4] 保留：保留位
将 0 写入这些位。

[bit3:0] BTMRST[3:0]：基本计时器的复位控制
这些位控制四个基本计时器单元的复位。以下为各位与基本计时器通道的对应关系：

bit0 - BTMRST0：基本计时器通道 0、1、2、3

bit1 - BTMRST1：基本计时器通道 4、5、6、7

bit2 - BTMRST2：基本计时器通道 8、9、10、11

bit3 - BTMRST3：基本计时器通道 12、13、14、15

如相关位设置为 1，对应的基本计时器通道单元变为复位状态，基本计时器停止运行，且寄存器设置初始化。对于未安装相关基本计时器通道的产品，不得变更其相关位的初始值。释放复位状态需再次将此位设置为 0。

位	描述
0	释放相关位对应的基本计时器通道的复位。（初始值）
1	向相关位对应的基本计时器通道发出复位信号。

4.5 外设时钟控制寄存器 2 (CKEN2)

本节说明外设时钟控制寄存器 2 (CKEN2)。

位	31	30	29	28	27	26	25	24
字段	保留							
属性	-							
Initial value	-							

位	23	22	21	20	19	18	17	16
字段	保留			PCRCCK	Reserved	CECCK	Reserved	LCDCCK
属性	-			R/W	-	R/W	-	R/W
初始值	-			1	-	1	-	1

位	15	14	13	12	11	10	9	8
字段	I2SCCK [1:0]		ICCCK [1:0]		保留		保留	
属性	R/W		R/W		-		-	
初始值	11		11		-		-	

位	7	6	5	4	3	2	1	0
字段	保留		CANCK[1:0]		保留		USBCK [1:0]	
属性	-		R/W		-		R/W	
初始值	-		11		-		00	

[bit31:21] 保留：保留位

将 0 写入这些位。

[bit20] PCRCCK：对可编程 CRC 提供和门控操作时钟的设置

此位控制针对可编程 CRC 功能的操作时钟的提供及门控可编程。此位设置为 1 时，向可编程 CRC 单元提供总线时钟，以使用可编程 CRC 功能。对于未安装相关可编程 CRC 单元的产品，不得变更其相关位的初始值。

此位设置为 0 时，对门控向可编程 CRC 单元输入总线时钟。门控总线时钟输入时，不能使用可编程 CRC 功能。

位	描述
0	门控向可编程 CRC 单元输入总线时钟。
1	向可编程 CRC 提供总线时钟。（初始值）

[bit19] 保留：保留位

将 0 写入此位。

[bit18] CECCK：对 HDMI-CEC/遥控接收提供和门控操作时钟的设置

此位控制针对 HDMI-CEC/遥控接收的操作时钟的提供及门控遥控接收。

相关位设置为 "1" 时，向 HDMI-CEC/遥控接收功能提供总线时钟。对于未安装相关 HDMI-CEC/遥控接收通道的产品，禁止变更其相关位的初始值。
相关位设置为 "0" 时，门控向 HDMI-CEC/遥控接收功能输入总线时钟。门控总线时钟输入时，不能使用 HDMI-CEC/遥控接收功能。

位	描述
0	门控向 HDMI-CEC/遥控接收功能输入总线时钟。
1	向 HDMI-CEC/遥控接收器功能提供总线时钟。（初始值）

[bit17] 保留：保留位
将 0 写入此位。

[bit16] LCDCCK：对 LCD 控制器提供和门控操作时钟的设置
此位控制针对 LCDC 功能的操作时钟的提供及门控。

此位设置为 1 时，向 LCDC 单元提供总线时钟，以使用 LCD 功能。

将此位设置为 0 时，门控向 LCDC 单元输入总线时钟。门控总线时钟输入时，不能使用 LCD 功能。

位	描述
0	门控向 LCDC 单元输入总线时钟。
1	向 LCDC 提供总线时钟。（初始值）

[bit15:14] I2SCCK[1:0]：对 MFSI2S 提供和门控操作时钟的设置
这些位控制针对 MFSI2S 的操作时钟的提供及门控。以下为各位与 MFSI2S 通道的对应关系：

[TYPE 2-M0+ 产品]

bit14– I2SCCK0：MFSI2S 串口通道 5

bit15– I2SCCK1：MFSI2S 串口通道 6

[TYPE 3-M0+ 产品]

bit14– I2SCCK0：MFSI2S 串口通道 4

bit15– I2SCCK1：MFSI2S 串口通道 6

相关位设置为 1 时，向对应的 MFSI2S 通道提供总线时钟，以使用 MFSI2S 功能。对于未安装相关 MFSI2S 通道的产品，禁止变更其相关位的初始值。

相关位设置为 0 时，门控向对应 MFSI2S 通道输入总线时钟。门控总线时钟输入时，不能使用对应通道的 MFSI2S 功能。

位	描述
0	门控向对应相关位的 MFSI2S 通道输入总线时钟。
1	向相关位对应的 MFSI2S 通道提供总线时钟。（初始值）

[bit13:12] ICCCK[1:0]：对智能卡接口提供和门控操作时钟的设置
这些位控制针对智能卡接口的操作时钟的提供及门控。以下为各位与智能卡接口通道的对应关系：

bit12– ICCCK0：智能卡接口通道 0

bit13– ICCCK1：智能卡接口通道 1

相关位设置为 1 时，向对应的智能卡接口通道提供总线时钟，以使用智能卡接口功能。对于未安装相关智能卡接口通道的产品，禁止变更其相关位的初始值。

相关位设置为 0 时，门控向对应智能卡接口通道输入总线时钟。门控总线时钟输入时，不能使用对应通道的智能卡接口功能。

位	描述
0	门控向对应相关位的智能卡接口通道输入总线时钟。
1	向相关位对应的智能卡接口通道提供总线时钟。（初始值）

[bit11:6] 保留：保留位
将 0 写入这些位。

[bit5:4] CANCK [1:0]：设置对 CAN 控制器的时钟提供和门控
这些位控制对 CAN 控制器的总线时钟（基本时钟）的提供和门控。以下为各位与 CAN 控制器通道的对应关系：

bit4 - CANCK0：CAN 控制器通道 0

bit5 - CANCK1：CAN 控制器通道 1

相关位设置为 1 时，向对应的 CAN 控制器通道提供总线时钟，以使用 CAN 控制器功能。对于未安装相关 CAN 控制器通道的产品，不得变更其相关位的初始值。

相关位设置为 0 时，门控向对应 CAN 控制器通道输入总线时钟。门控总线时钟输入时，不能使用对应通道的 CAN 控制器功能。

位	描述
0	门控向相关位对应的 CAN 控制器通道输入总线时钟。（初始值：适用于未安装 CAN 控制器的产品）
1	向相关位对应的 CAN 控制器通道提供总线时钟。（初始值：适用于安装了 CAN 控制器的产品）

[bit3:2] 保留：保留位
将 0 写入这些位。

[bit1:0] USBCK[1:0]：对操作时钟的提供和 USB（从机/主机）门控设置
这些位控制针对 USB(从机/主机)的操作时钟的提供及门控。以下为各位与 USB 通道的对应关系：

bit0 - USBCK0：USB 通道 0

bit1 - USBCK1：USB 通道 1

相关位设置为 1 时，向对应的 USB 通道提供总线时钟，以使用 USB 功能。对于未安装相关 USB 通道的产品，禁止变更其相关位的初始值。

相关位设置为 0 时，门控向对应 USB 通道输入总线时钟。门控总线时钟输入时，不能使用对应通道的 USB 功能。

位	描述
0	门控向相关位对应的 USB 通道输入总线时钟。（初始值）
1	向相关位对应的 USB 通道提供总线时钟。

4.6 外设功能复位控制寄存器 2 (MRST2)

本节说明外设功能复位控制寄存器 2 (MRST2)。

位	31	30	29	28	27	26	25	24
字段	保留							
属性	-							
初始值	-							

bit	23	22	21	20	19	18	17	16
字段	保留			PCRCRST	保留	CECRST	保留	LCDCRST
属性	-			R/W	-	R/W	-	R/W
初始值	-			0	-	0	-	0

位	15	14	13	12	11	10	9	8
字段	I2SCRST [1:0]		ICCRST [1:0]		保留		保留	
属性	R/W		R/W		-		-	
初始值	00		00		-		-	

位	7	6	5	4	3	2	1	0
字段	保留		CANRST[1:0]		保留		USBRST [1:0]	
属性	-		R/W		-		R/W	
初始值	-		00		-		00	

[bit31:21] 保留：保留位

将 0 写入这些位。

[bit20] PCRCRST：可编程 CRC 的复位控制

此位控制可编程 CRC 单元的复位。如此位设置为 1，可编程 CRC 变为复位状态，可编程 CRC 停止运行，且寄存器设置初始化。对于未安装可编程 CRC 的产品，勿将此位设置为 1。释放上述复位需再次将此位设置为 0。

位	描述
0	解除可编程 CRC 的复位。（初始值）
1	向可编程 CRC 发出复位信号。

[bit19] 保留：保留位

将 0 写入此位。

[bit18] CECRST：HDMI-CEC/遥控接收的复位控制

此位控制 HDMI-CEC/遥控接收的复位。

如相关位设置为 1，HDMI-CEC/遥控接收变为复位状态，HDMI-CEC/遥控接收停止运行，且寄存器设置初始化。

释放复位状态需再次将此位设置为 0。

位	描述
0	释放 HDMI-CEC/遥控接收的复位。（初始值）
1	向 HDMI-CEC/遥控接收发出复位信号。

[bit17] 保留：保留位

将 0 写入此位。

[bit16] LCDCRST：LCD 控制器的复位控制

此位控制 LCDC 功能的复位。

此位设置为 1 时，LCDC 变为复位状态，LCDC 停止运行，且寄存器设置初始化。对于未安装 LCDC 的产品，勿将此位设置为 1。释放上述复位需再次将此位设置为 0。

位	描述
0	释放 LCDC 复位。（初始值）
1	向 LCDC 发出复位信号。

[bit15:14] I2SCRST [1:0]：MFSI2S 的复位控制

这些位控制 MFSI2S 各通道的复位。以下为各位与 MFSI2S 通道的对应关系：

[TYPE 2-M0+ 产品]

bit14– I2SCRST0：MFSI2S 串口通道 5

bit15– I2SCRST1：MFSI2S 串口通道 6

[TYPE 3-M0+ 产品]

bit14– I2SCRST0：MFSI2S 串口通道 4

bit15– I2SCRST1：MFSI2S 串口通道 6

如相关位设置为 1，对应的 MFSI2S 通道变为复位状态，MFSI2S 停止运行，且寄存器设置初始化。对于未安装相关 MFSI2S 通道的产品，不得变更此位的初始值。释放复位状态需再次将此位设置为 0。

位	描述
0	释放相关位对应的 MFSI2S 通道的复位。（初始值）
1	向相关位对应的 MFSI2S 通道发出复位信号。

[bit13:12] ICCRST[1:0]：智能卡接口的复位控制

这些位控制智能卡接口各通道的复位。以下为各位与智能卡接口通道的对应关系：

bit12– ICCRST0：智能卡接口通道 0

bit13– ICCRST1：智能卡接口通道 1

如相关位设置为 1，对应的智能卡接口通道变为复位状态，智能卡接口停止运行，且寄存器设置初始化。对于未安装相关智能卡接口通道的产品，不得变更此位的初始值。释放复位状态需再次将此位设置为 0。

位	描述
0	释放相关位对应的智能卡接口通道的复位。（初始值）
1	向相关位对应的智能卡接口通道发出复位信号。

[bit11:6] 保留：保留位
将 0 写入这些位。

[bit5:4] CANRST [1:0]：CAN 控制器的复位控制
这些位控制各 CAN 控制器通道单元的复位。以下为各位与 CAN 控制器通道的对应关系：

bit4 - CANRST0：CAN 控制器通道 0

bit5 - CANRST1：CAN 控制器通道 1

如相关位设置为 1，对应的 CAN 控制器通道变为复位状态，CAN 控制器停止运行，且寄存器设置初始化。对于未安装 CAN 控制器通道的产品，不得变更此位的初始值。释放复位状态需再次将此位设置为 0。

位	描述
0	释放相关位对应的 CAN 控制器通道的复位。（初始值）
1	向相关位对应的 CAN 控制器通道发出复位信号。

[bit3:2] 保留：保留位
将 0 写入这些位。

[bit1:0] USBRST [1:0]：USB（从机/主机）的复位控制
这些位控制 USB（从机/主机）各通道的复位。以下为各位与 USB 通道的对应关系：

bit0 - USBRST0：USB 通道 0

bit1 - USBRST1：USB 通道 1

如相关位设置为 1，对应的 USB 通道变为复位状态，USB 停止运行，且寄存器设置初始化。对于未安装相关 USB 通道的产品，不得变更此位的初始值。释放复位状态需再次将此位设置为 0。

位	描述
0	释放相关位对应的 USB 通道的复位。（初始值）
1	向相关位对应的 USB 通道发出复位信号。

5. 外设时钟门控功能使用注意事项

本节说明外设时钟门控功能使用注意事项。

概述

■ 时钟提供停止的外设功能控制

寄存器读写访问时钟提供已停止的外设功能均得不到保证。因为读取值未定义，写入操作被禁止。
外设时钟被门控时，可通过控制外设功能复位控制寄存器（MRST0、MRST1 及 MRST2）复位内部状态。

■ 外设时钟设置组合

确保通过外设时钟寄存器（CKEN0、CKEN1 及 CKEN2）将所有目标外设功能设置为时钟提供侧，以运行两种或多种外设功能组合。例如，当通过 A/D 转换器的计时器触发信号选择使用基本计时器时，可通过外设时钟控制寄存器（CKEN0 及 CKEN1）分别将所用 A/D 转换器相关单元以及基本计时器的相关通道设置为时钟提供侧。

■ 外设时钟设置的初始化条件

通过以下复位可初始化外设时钟门控功能。发出以下复位命令后，务必重新配置外设时钟门控功能。
关于以下复位命令的详细信息，参见“复位”一章。

- 上电复位 (PONR)
- 低压检测复位 (LVDH)
- INITX 引脚输入 (INITX)
- 软件看门狗复位 (SWDGR)
- 硬件看门狗复位 (HWDGR)
- 时钟故障检测复位 (CSVR)
- 异常频率检测复位 (FCSR)
- 软件复位 (SRST)
- APB1 总线复位 (APBC1_PSR)
- 深度待机转换复位 (DSTR)

多功能串口

■ LIN 同步字段检测：LSYN

LIN 总线接口模式下使用输入捕捉 (ICU) 时，分别执行对应多功能计时器（输入捕捉）操作时钟提供的设置以及多功能串口外设时钟的设置。关于多功能串口与输入捕捉之间的连接，参见《FM0+ 外设手册》“I/O 口”一章中的“扩展引脚功能设置寄存器(EPFR) ”一节。

基本计时器

■ 基本计时器的时钟设置单元

基本计时器的外设时钟控制在 Table 5-1 所述四通道单元内执行。

Table 5-1 外设时钟门控设置与基本计时器通道的对应关系

外设时钟控制寄存器 (CKEN1) 的设置位	目标通道
bit 0	基本计时器 ch3, ch2, ch1, ch0
bit 1	基本计时器 ch7, ch6, ch5, ch4
bit 2	基本计时器 ch11, ch10, ch9, ch8
bit 3	基本计时器 ch15, ch14, ch13, ch12

多功能计时器

FRT 选择寄存器

使用以下 FRT 选择功能时，设置源侧 FRT 安装在提供侧的多功能计时器单元的操作时钟。

- OCU 连接 FRT 选择寄存器 (OCFS)
- ICU 连接 FRT 选择寄存器 (ICFS)
- ADC S 启动比较连接 FRT 选择寄存器 (ADCMP)

PPG

■ PPG 的时钟控制

PPG 输入时钟的控制与多功能计时器输入时钟的设置同步。关于 PPG 通道数量及多功能计时器单元数量，参见 Table 5-2。

Table 5-2 多功能计时器及 PPG 输入时钟控制

多功能计时器单元号	PPG 通道号
单元 0	通道 0,1,2,3,4,5,6,7
单元 1	通道 8,9,10,11,12,13,14,15
单元 2	通道 16,17,18,19,20,21,22,23
单元 3	通道 24,25,26,27,28,29,30,31

USB（从机/主机）

■ 时钟控制目标

不能使用外设时钟控制寄存器 2 (CKEN2) 的 USBCK 位控制 USB 通讯的时钟门控及提供。使用 USB 时钟控制寄存器 (UCCR) 的 UCEN 位控制 USB 通讯时钟。详细信息参见《FM0+ 外设手册》 “I/O 口” 通讯宏部分的 USB 时钟生成一章。

A/D 转换器

■ A/D 计时器触发信号选择

基本计时器用作 A/D 转换器的启动触发信号时，将选择基本计时器通道的操作时钟设置到提供侧。

GPIO/快速 GPIO

■ 总线时钟门控时的限制

如 Table 5-3 所示，GPIO/快速 GPIO 受门控时，I/O 口的某些功能不可使用。
务必确认使用条件并执行 GPIO/快速 GPIO 的总线时钟控制。
关于 I/O 口功能的详细信息，参见 I/O 口及快速 GPIO 一章。

Table5-3 GPIO/快速 GPIO 时钟门控时的限制

限制	总线时钟状态	
	提供*	门控*
I/O 口功能输入电平读取（PDIR/FPDIR/M_FPDIR 寄存器读取）	有	禁止
I/O 口功能输出电平转换及状态确认（PDOR/FPDOR/M_FPDOR 寄存器读取/写入）	有	禁止
I/O 口模式转换（PFR、PCR、DDR、ADE、SPSR、EPFR、and PZR、FPOER 寄存器的设置变更）	有	禁止
外设功能操作（信号输入及输出）	有	有
外部中断/NMI 控制	有	有
复位输入 (INITX)	有	有
从深度待机模式（WKUP 引脚输入）唤醒	有	有

*：有：可以使用；禁止：不能使用。

HDMI-CEC/遥控接收器

■ 时钟控制目标

不能使用外设时钟控制寄存器 2 (CKEN2) 的 CECCK 位控制 HDMI-CEC/遥控接收的副时钟门控及提供。使用副时钟控制寄存器 (RCK_CTL) 的 CECCKE 位控制 HDMI-CEC/遥控接收的副时钟。详细信息参见低功耗模式一章。

第 2-3 章：高速 CR 调节

本章说明高速 **CR** 调节功能。

1. 高速 CR 调节功能概述
2. 高速 CR 调节功能配置及框图
3. 高速 CR 调节功能操作
4. 高速 CR 调节功能设置步骤示例
5. 高速 CR 调节功能寄存器表
6. 高速 CR 调节功能使用注意事项

代码：9BFCRTRIM_FM0-C03.0

1. 高速 CR 调节功能概述

本节说明高速 CR 振荡器的频率调节功能。

由于工艺不同，本设备所用高速 CR 振荡器的频率精确度有一定的波动范围。 通过配置调节功能，可以减小不同工艺和温度变化造成的频率精确度的波动幅度。

高速 CR 调节功能包括频率调节设置单元和温度调节设置单元。

频率调节设置单元具有以下功能：

- 可以通过将调节值写入高速 CR 振荡频率调节寄存器 (MCR_FTRM)，配置高速 CR 频率调节。
- 使用基本计时器的 ch.0，可以通过指定周期内的计数值计算写入频率调节寄存器中的设置值。

温度调节设置单元具有以下功能：

可以通过将调节值写入高速 CR 振荡温度调节寄存器 (MCR_TTRM)，配置高速 CR 温度补偿。

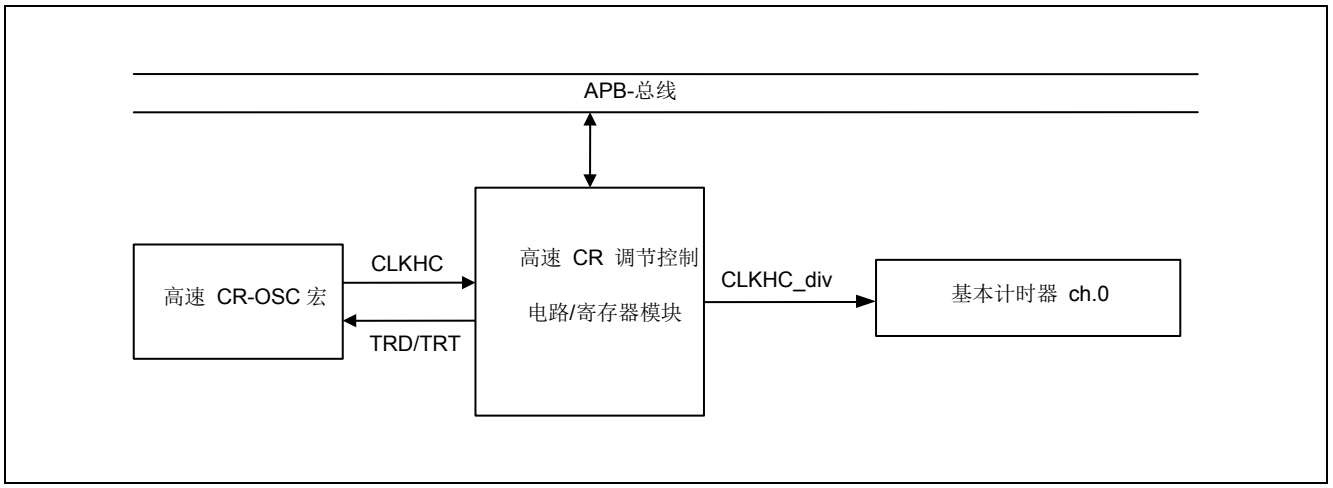
有关高速 CR 频率精确度，参见所用产品《数据手册》中的电气特性。

2. 高速 CR 调节功能配置及框图

本章说明高速 CR 振荡器频率调节功能配置及框图。

Figure 2-1 所示为高速 CR 频率调节功能框图。

Figure 2-1 高速 CR 振荡器计时电路框图



配置

■ 高速 CR OSC 模块

高速 CR 时钟模块输出 CLKHC（高速 CR 时钟）。

此外，可通过高速 CR 振荡频率调节寄存器 (MCR_FTRM) 中的 TRD 位和高速 CR 振荡温度调节寄存器 (MCR_TTRM) 的 TRT 位执行频率调节。

■ 高速 CR 调节控制电路和寄存器单元。

用于调节高速 CR 的控制电路和寄存器。

此外，高速 CR 振荡分频设置寄存器 (MCR_PSR) 中 CSR 位所设比率所分频的高速 CR 时钟 (CLKHC_div) 输出至基本计时器 ch.0。

■ 基本计时器

此单元在设置之前对频率进行计数，以计算高速 CR 频率调节数据。

注意事项：

- *有关时钟的定义，参见 "时钟" 一章。*

3. 高速 CR 调节功能操作

本节说明高速 CR 振荡器的频率调节功能相关的操作。

高速 CR 振荡频率调节功能操作

■ 频率调节设置

设置进程将调节数据值写入高速 CR 振荡频率调节寄存器 (MCR_FTRM)，纠正不同工艺造成的高速 CR 时钟精确度偏差。

■ 温度调节设置

设置进程将调节数据值写入高速 CR 振荡温度调节寄存器 (MCR_TTRM)，纠正温度变化造成的高速 CR 时钟精确度偏差。

■ 寄存器锁定功能

为高速 CR 振荡频率调节寄存器 (MCR_FTRM) 和高速 CR 振荡温度调节寄存器 (MCR_TTRM) 提供写入保护功能，避免在系统失去控制时擅自重写寄存器。

■ 调节数据采集

可通过以下三种方法之一采集写入高速 CR 振荡频率调节寄存器 (MCR_FTRM) 的数据：

- 使用储存在闪存内 "CR 调节" 区中的工厂预设值。解除复位后，闪存内 CR 调节区中的值储存在 CR 调节数据镜像寄存器 (CRTRMM) 中。 使用 CR 调节数据镜像寄存器 (CRTRMM) 的 TRMM 位，采集写入高速 CR 振荡频率调节寄存器 (MCR_FTRM) 的数据。
- 使用基本计时器在一定时期内的计数值独自计算高速 CR 振荡频率调节寄存器的设置值。
- 输出高速 CR 时钟至外部引脚，监控波形调节频率并独自计算高速 CR 振荡频率调节寄存器的设置值。

可通过以下方法采集写入高速 CR 振荡温度调节寄存器 (MCR_TTRM) 的数据：

- 使用储存在闪存内 CR 调节区中的出厂预设值。

释放复位后，闪存内 CR 调节区中的值储存在 CR 调节数据镜像寄存器 (CRTRMM) 中。数据写入高速 CR 振荡温度调节寄存器 (MCR_TTRM) 时，使用 CR 调节数据镜像寄存器 (CRTRMM) 的 TTRM 位，。

注意事项：

- *擦除闪存时，也擦除了存储器中的 "CR 调节" 区。 如果使用 "CR 调节" 区中的值，在擦除闪存前将数据保存在其他区（比如 RAM）中，或仅擦除 "CR 调节" 区之外的扇区。*
- *有关 "CR 调节" 区的地址，参见所用产品的《闪存编程手册》。*

4. 高速 CR 调节功能设置步骤示例

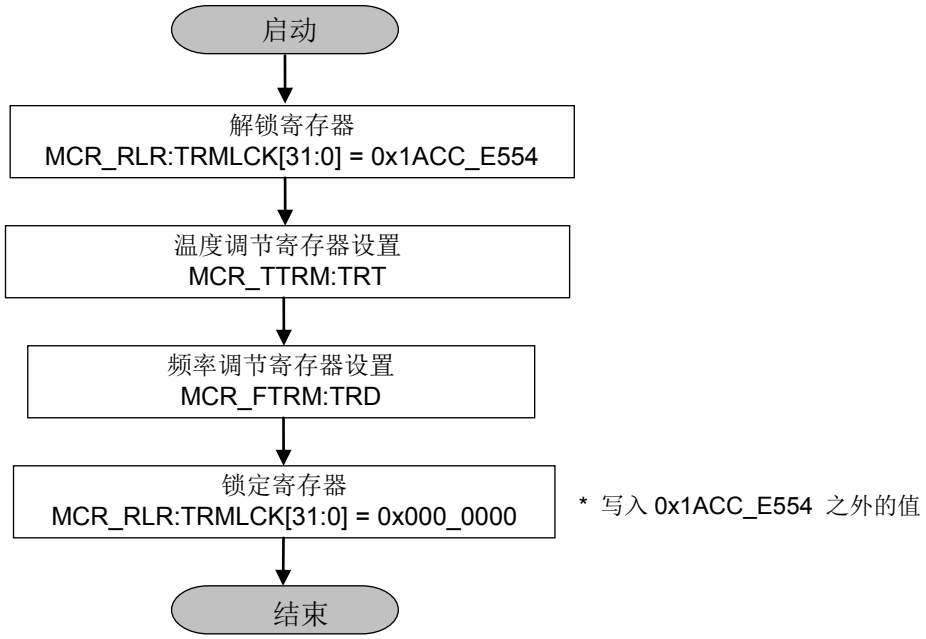
本节说明高速 CR 振荡器频率调节功能设置示例。

频率调节设置

按 Figure 4-1 所示的步骤设置频率调节。

1. 将 "0x1ACCE554" 写入高速 CR 频率寄存器写入保护寄存器 (MCR_RLR) 的 TRMLCK[31:0] 位，解锁频率调节设置寄存器 (MCR_FTRM) / 温度调节设置寄存器 (MCR_TTRM) 。
2. 设置温度调节寄存器 (MCR_TTRM) 的 TRT 位调节数据。
3. 设置频率调节设置寄存器 (MCR_FTRM) 的 TRD 位。
4. 将除 "0x1ACCE554" 之外的值写入高速 CR 振荡寄存器写入保护寄存器 (MCR_RLR) 的 TRMCLK[31:0] 位，以锁定频率调节设置寄存器 (MCR_FTRM) / 温度调节设置寄存器 (MCR_TTRM)。

Figure 4-1 频率/温度调节设置



频率调节数据采集示例

采集闪存内 "CR 调节" 区的数据时；

读取闪存内 "CR 调节" 区并取得数据。
将采集值写入高速 CR 振荡频率调节设置寄存器 (MCR_FTRM) 的 TRD 位。

如何计算频率调节数据

下文说明如何计算高速 CR 振荡的调节数据。

1. 使目标振荡频率设定为 Ftgt[MHz]，其周期为 Ttgt[ns]。 使 Xtrm_coarse 和 Xtrm_fine 分别为高速 CR 振荡频率调节设置寄存器当时的 TRD[9:5] 位值和 TRD[4:0] 位值。
2. TRD[4:0] 位设为 “0b00000”。
3. 使得 TRD[4:0] 位设为 “0b00000” 时，Xtrm_coarse 为 Xtrmmin_coarse。 使得 Tmax_coarse[sec] 为当时的周期。
4. 使得 TRD[9:5] 位设为 “0b11111” 时，Xtrm_coarse 为 Xtrmmax_coarse。 使得 Tmin_coarse[sec] 为当时的周期。
5. 用下述方程式计算出 TRD[9:5] 设置值，Xtrm_coarse，给定值大于目标振荡周期，Ttgt。

$$Xtrm_coarse = \frac{Ttgt - \frac{T\max_coarse - T\min_corse}{31} - T\max_coarse}{\frac{T\min_coarse - T\max_coarse}{31}}$$

*：小数四舍五入。

6. 将得到的 Xtrm_coarse 设置到 TRD[9:5] 位。
7. 设置 TRD 位后，确认高速 CR 时钟、F_{CRH} 为 Ftgt 或更小值。 如果 F_{CRH} 超过 Ftgt，Xtrm_coarse 减去 “1” 然后返回第 6 步，当 F_{CRH} 为 Ftgt 或以下时，转到第 8 步。
8. 使得 TRD[4:0] 位被设置为 “0b00000” 时，该值为 Xtrimmin_fine。 使得 Tmax_fine[sec] 为当时的周期。
9. 使得 TRD[4:0] 位被设置为 “0b11111” 时，该值为 Xtrimmax_fine。 使得 Tmin_fine[sec] 为当时的周期。
10. 使用下述方程式计算 TRD[4:0] 设置值，Xtrm_fine ，给定目标振荡周期，Ttgt。

$$Xtrm_fine = \frac{Ttgt - \frac{T\max_fine - T\min_fine}{31} - T\max_fine}{\frac{T\min_fine - T\max_fine}{31}}$$

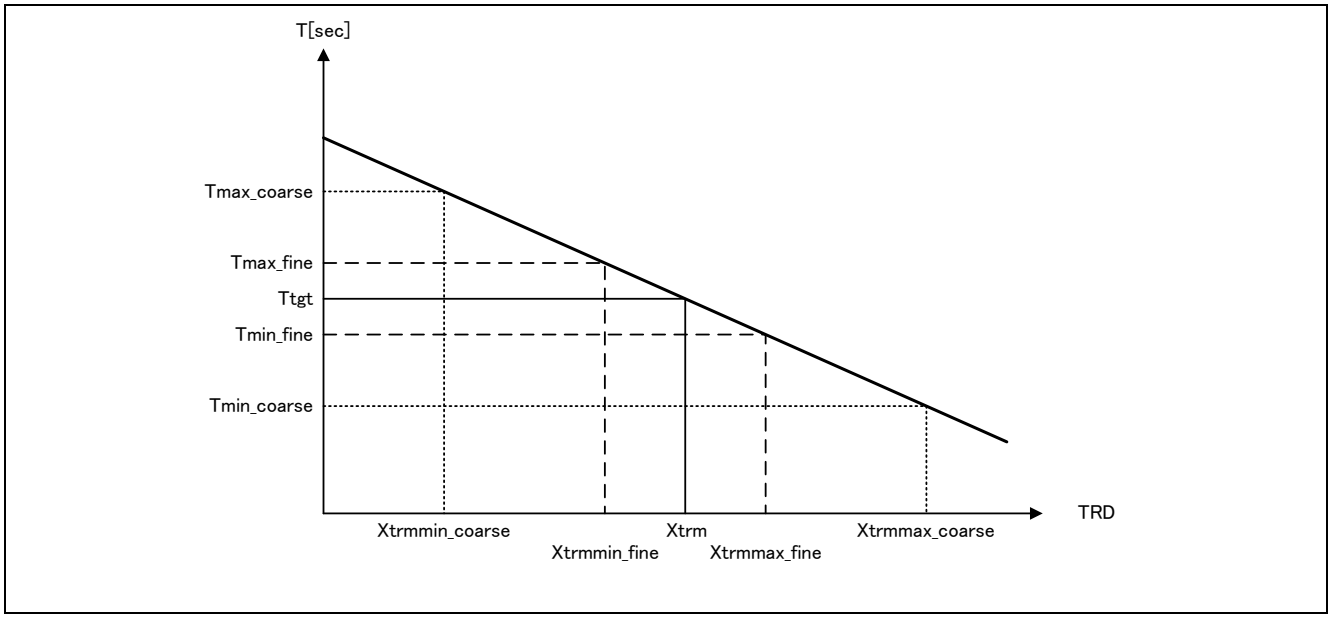
*：小数四舍五入。

11. 将得到的 Xtrm_fine 设置到 TRD[4:0] 位。
12. 设置 TRD 位后，确认高速 CR 时钟、F_{CRH} 为 Ftgt 或更大值，且在 CR 时钟振荡频率规格值内。 如果 F_{CRH} 超过规格值，从 Xtrm_fine 减去 “1” 并返回第 11 步。 此外，如果 F_{CRH} 小于 Ftgt，在 Xtrm_fine 中加上 “1” 并返回第 11 步。当该值在规格值之内时，调节数据计算完成。

注意事项：

- *有关高速 CR 时钟振荡频率的规格，参见所使用产品的《数据手册》。*

Figure 4-2 高速 CR 时钟调节方法



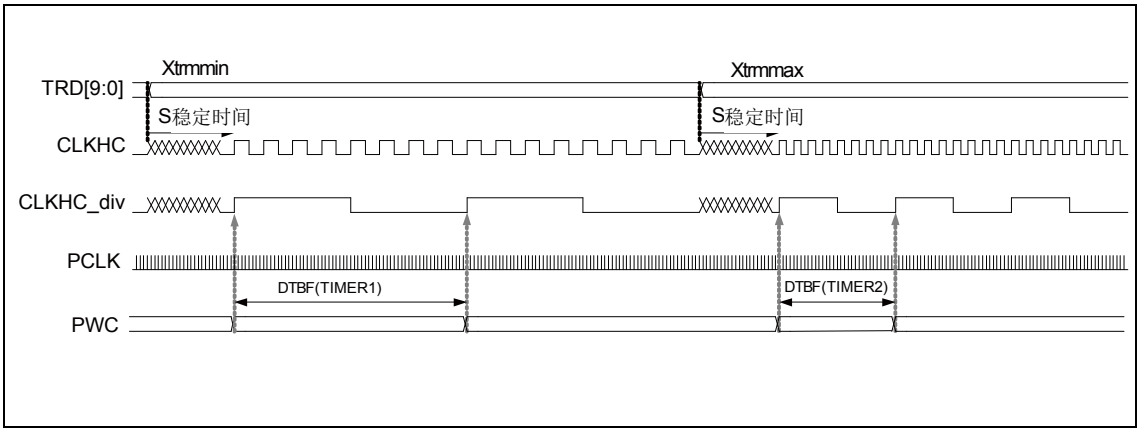
注意事项：

- *有关如何测量 Tmin_coarse/fine 和 Tmax_coarse/fine，参见 "使用基本计时器进行调节数据采集的示例 ".*

使用基本计时器进行调节数据采集的示例

Figure 4-3 所示为高速 CR 振荡和调节进程时间图。

Figure 4-3 使用基本计时器的高速 CR 振荡和调节进程时间图



通过将主振荡时钟 (CLKMO) 设置为主控时钟（测量基准时钟）运行基本计时器。
当设置 Xtrmmin 或 Xtrmmax 时，激活高速 CR 分频时钟 (CLKHC_div) 上升沿上的触发信号，读取当时的基本计时器值并执行以下计算。

Tmax = (TIMER1×PCLK) / DIV

Tmin = (TIMER2×PCLK) / DIV

- TIMER1、TIMER2：　基本计时器计数值 (PWC)
- PCLK：　APB1 总线时钟
- DIV：　通过分频设置寄存器 (MCR_PSR) 的 CSR 位设置分频比

示例：当 PCLK = 40MHz (25ns) 时，分频比 = 1/8，TIMER1 = 100，

Tmax = (100 × 25 ns) / 8 = 312.5 ns

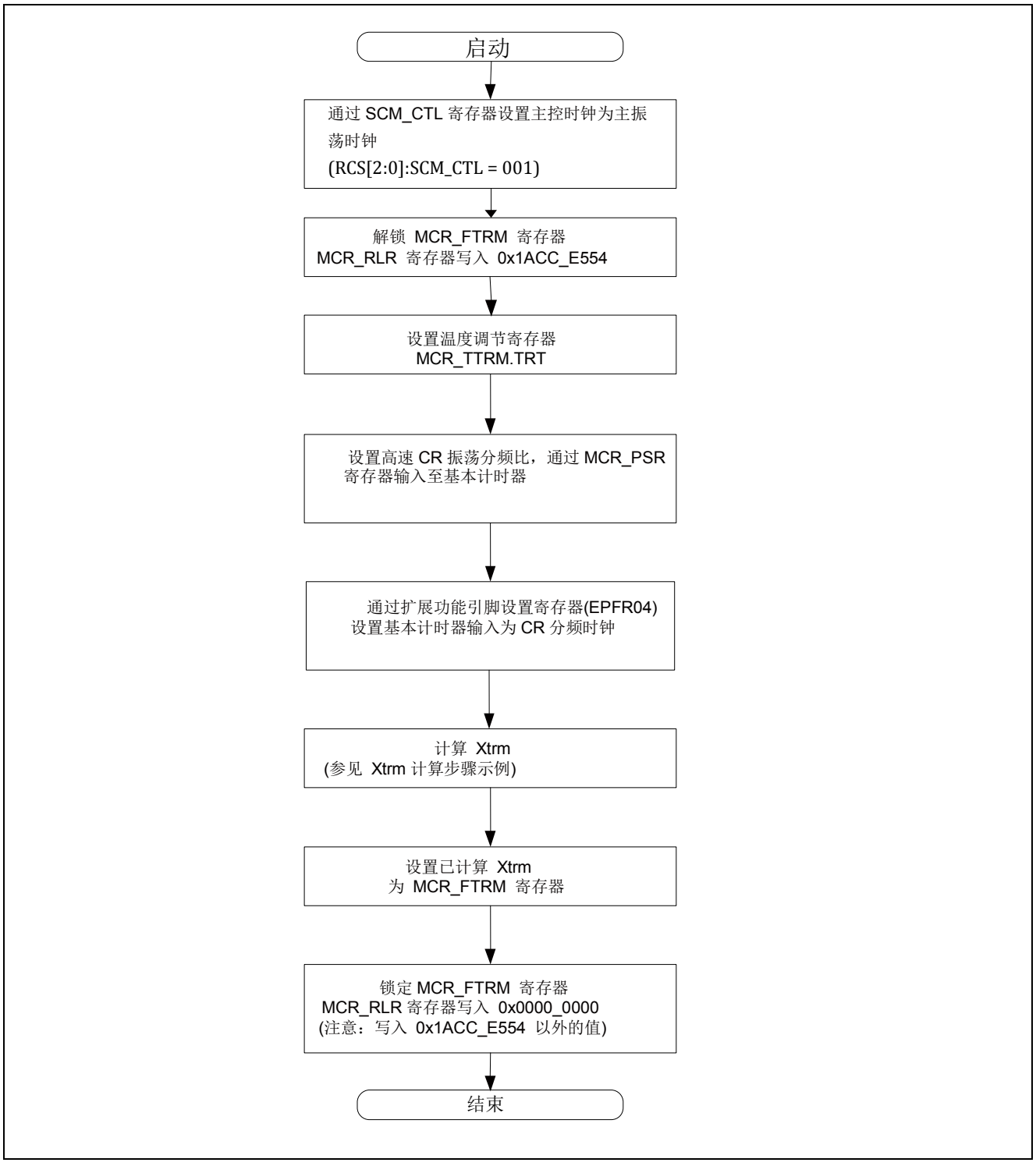
注意事项：

- *用于调节的基本计时器为 ch.0。*
  Figure 4-3 所示的 PCLK 为 APB1 总线时钟。
  此时，选择主时钟作为 PCLK 主振荡。

频率调节步骤示例

Figure 4-4 所示为高速 CR 振荡调节步骤示例。

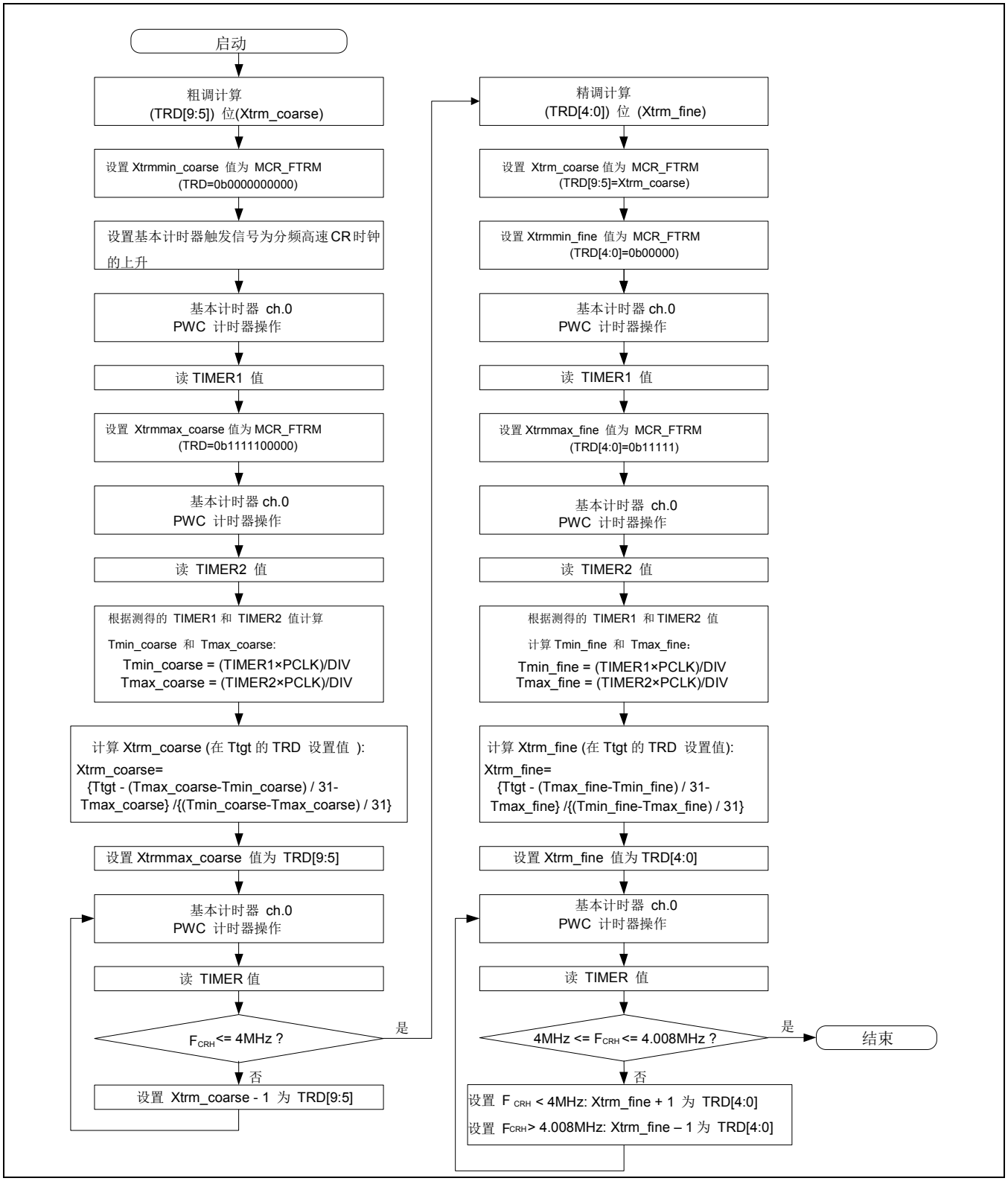
Figure 4-4 高速 CR 振荡调节步骤示例



Xtrm 计算步骤示例（4MHz 振荡器）

Figure 4-5 所示为 4MHz 高速 CR 振荡器的 Xtrm 计算步骤示例。分两级以粗调和精调执行频率调节。

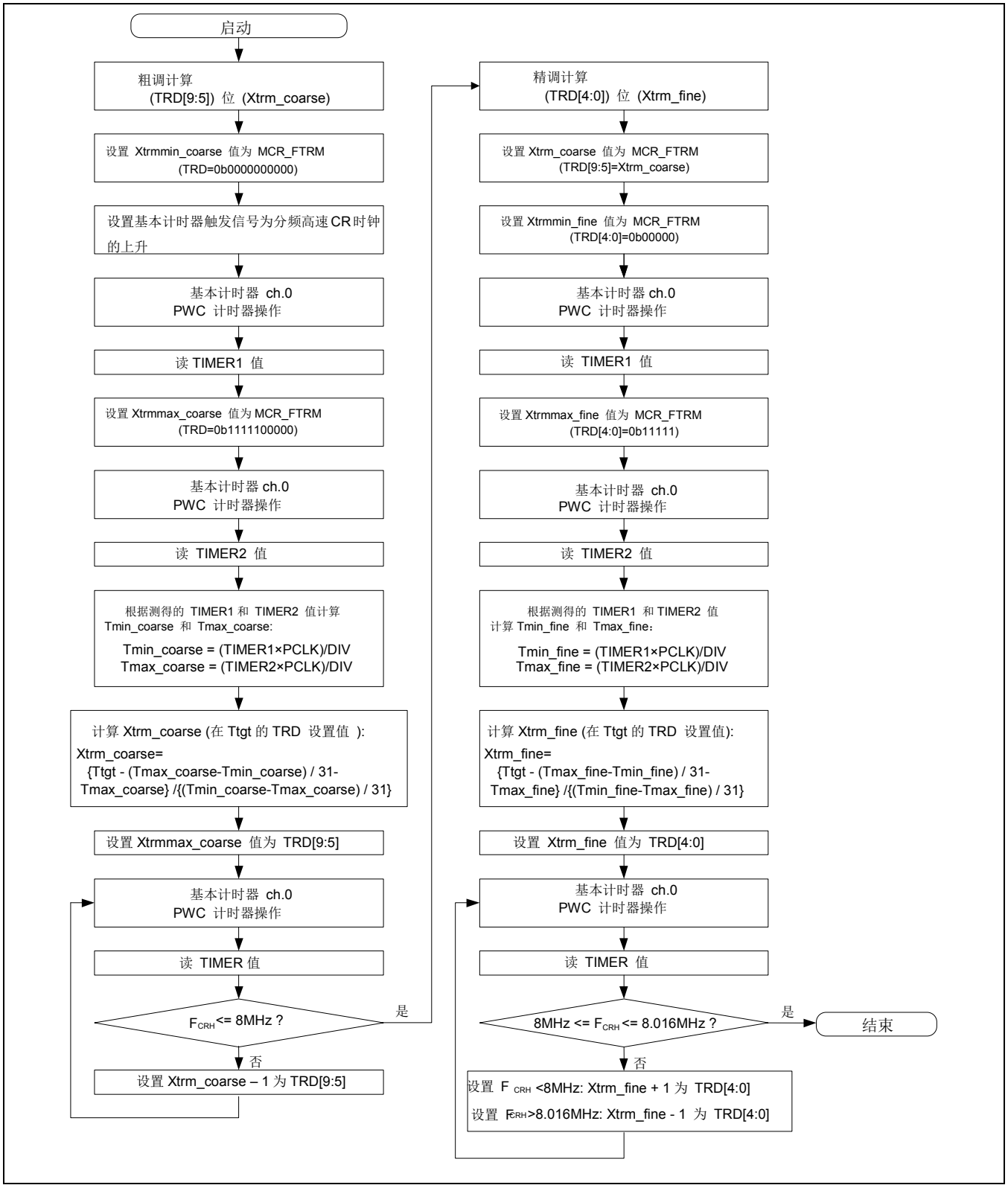
Figure 4-5 Xtrm 计算步骤示例 (4MHz)



Xtrm 计算步骤示例（8MHz 振荡器）

Figure 4-6 所示为 8MHz 高速 CR 振荡器的 Xtrm 计算步骤示例。分两级以粗调和精调执行频率调节。

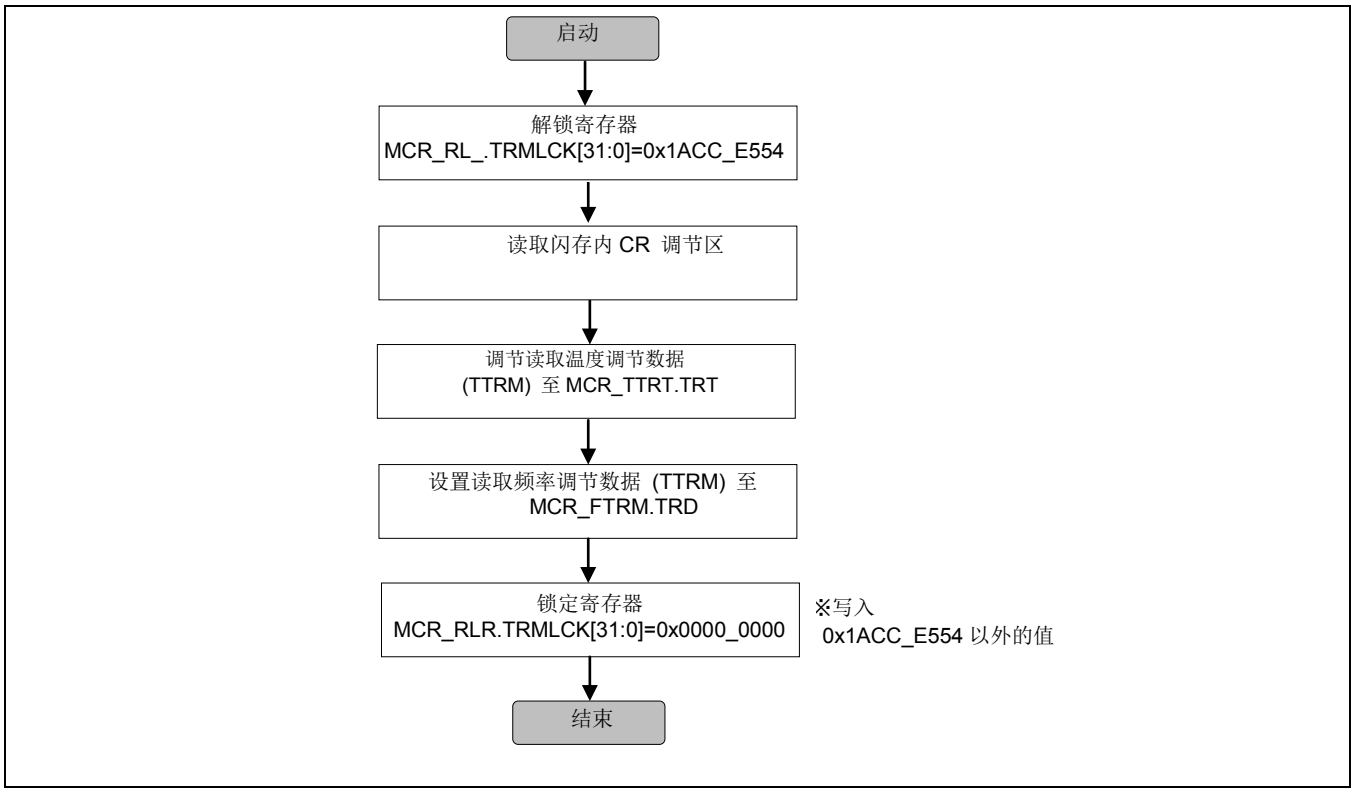
Figure 4-6Xtrm 计算步骤示例 (8MHz)



使用闪存内 "CR 调节" 区储存数据的步骤示例

Figure 4-7 所示为读取闪存内 "CR 调节" 区所储存调节数据的步骤示例，并在高速 CR 振荡频率调节寄存器中设置该值。

Figure 4-7 使用 "CR 调节" 区储存数据的步骤示例



注意事项：

- *有关 CR 调节区的地址，参见所用产品的《闪存编程手册》 。*

5. 高速 CR 调节功能寄存器表

以下列出并说明高速 CR 振荡器频率调节功能使用的寄存器。

Table 5-1 列出寄存器。

Table 5-1 寄存器表

缩写	寄存器名称	参考章节
MCR_PSR	高速 CR 振荡分频设置寄存器	**5.1**
MCR_FTRM	高速 CR 振荡频率调节寄存器	**5.2**
MCR_TTRM	高速 CR 振荡温度调节寄存器	**5.3**
MCR_RLR	高速 CR 振荡寄存器写保护寄存器	**5.4**

5.1 高速 CR 振荡分频设置寄存器 (MCR_PSR)

MCR_PSR 寄存器设置高速 CR 振荡分频比。
分频时钟可输入基本计时器。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	保留						CSR	
属性	-						R/W	
初始值	-						001	

寄存器功能

[bit7:3] 保留：保留位
这些位读为 "0b00000"。
写入时这些位设为 "0b00000"。

[bit2:0] CSR：高速 CR 振荡分频比设置位

bit2	bit1	bit0	描述
0	0	0	1/4
0	0	1	1/8 [初始值]
0	1	0	1/16
0	1	1	1/32
1	0	0	1/64
1	0	1	1/128
1	1	0	1/256
1	1	1	1/512

5.2 高速 CR 振荡频率调节寄存器 (MCR_FTRM)

MCR_FTRM 寄存器设置频率调节值。

寄存器配置

位	31															16
字段	保留															
属性	-															
初始值	-															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留						TRD[9:0]									
属性	-						R/W									
初始值	-						0111101111 （TYPE 1-M0+、TYPE 2-M0+） 1000000110（TYPE 3-M0+）									

寄存器功能

[bit31:10] 保留：保留位

这些位始终读为 "0"。

写入时这些位无效。

[bit9:0] TRD[9:0]：频率调节设置位

bit9:5	说明
写入时	这些位对高速 CR 振荡器频率进行粗调。 有关要设置的值，参见频率调节功能操作说明中的调节数据采集。 每次 ±1 设置时，这些位波动的频率阶跃如下： 大约 1.0% (TYPE1-M0+)，大约 3.3% (TYPE2-M0+)，大约 1.8% (TYPE3-M0+)
读取时	读取指定值。 初始值读取 "0b01111"。 （TYPE 1-M0+、TYPE 2-M0+） 初始值读取 "0b10000"。 （TYPE 3-M0+）

bit4:0	描述
写入时	这些位对高速 CR 振荡器频率进行细调。 有关要设置的值，参见频率调节功能操作说明中的调节数据采集。 每次 ±1 设置时，这些位波动的频率阶跃如下： 大约 0.2% (TYPE1-M0+)，大约 0.3% (TYPE2-M0+)，大约 0.1% (TYPE3-M0+)
读取时	读取指定值。 初始值读取 "0b01111"。 （TYPE 1-M0+、TYPE 2-M0+） 初始值读取 "0b00110"。 （TYPE 3-M0+）

注意事项：

- *软件复位不能初始化本寄存器。*
- *有关设置至 TRD 位的值，参见频率调节功能操作说明所示的调节数据采集。*

5.3 高速 CR 振荡温度调节设置寄存器 (MCR_TTRM)

MCR_TTRM 寄存器设置温度调节值。

TYPE 1-M0+ 和 TYPE 2-M0+ / TYPE 3-M0+ 的寄存器定义不同。

5.3.1 MCR_TTRM （TYPE 1-M0+）

寄存器配置

位	31															16
字段	保留															
属性	-															
初始值	-															

位	15											5	4	3	2	1	0
字段	保留												TRT[4:0]				
属性	-												R/W				
初始值	-												10000				

寄存器功能

[bit31:5] 保留：保留位

这些位始终读取 "0"。

写入时这些位无效。

[bit4:0] TRT[4:0]：温度调节设置位

bit4:0	描述
写入时	这些位对高速 CR 振荡器频率进行调节。 写入从闪存内温度调节位储存区读取的值。 有关温度调节位储存区，参见所用产品的《闪存编程手册》。
读取时	读取指定值。 初始值读取 0b10000。

注意事项：

- *软件复位不能初始化本寄存器。*

在获得频率调节数据之前，确保设置本寄存器。

5.3.2 MCR_TTRM（TYPE 2-M0+）

寄存器配置

位	31															16
字段	保留															
属性	-															
初始值	-															

位	15														6	5	4	3	2	1	0
字段	保留														TRT[6:0]						
属性	-														R/W						
初始值	-														0111111						

寄存器功能

[bit31:7] 保留：保留位
这些位始终读为 "0"。
写入时这些位无效。

[bit6:0] TRT[6:0]：温度调节设置位

bit4:0	描述
写入时	这些位对高速 CR 振荡器频率进行调节。 写入从闪存内温度调节位储存区读取的值。 有关温度调节位储存区，参见所用产品的《闪存编程手册》。
读取时	读取指定值。 初始值读取 0b0111111。

注意事项：

- *软件复位不能初始化本寄存器。*

在获得频率调节数据之前，确保设置本寄存器。

5.3.3 MCR_TTRM（TYPE 3-M0+）

寄存器配置

位	31															16
字段	保留															
属性	-															
初始值	-															

位	15															6	5	4	3	2	1	0
字段	保留															TRT[6:0]						
属性	-															R/W						
初始值	-															1111111						

寄存器功能

[bit31:7] 保留：保留位
这些位始终读取 "0"。
写入时这些位无效。

[bit6:0] TRT[6:0]：温度调节设置位

bit4:0	描述
写入时	这些位对高速 CR 振荡器频率进行调节。 写入从闪存内温度调节位储存区读取的值。 有关温度调节位储存区，参见所用产品的《闪存编程手册》。
读取时	读取指定值。 初始值读取 0b1111111。

注意事项：

- *软件复位不能初始化本寄存器。*
- *在获得频率调节数据之前，确保设置本寄存器。*

5.4 高速 CR 振荡寄存器写保护寄存器 (MCR_RLR)

MCR_RLR 寄存器控制频率调节寄存器 (MCR_FTRM) / 高速 CR 振荡温度调节寄存器 (MCR_TTRM) 的写入保护状态。

寄存器配置

位	31															16
字段	TRMLCK[31:16]															
属性	R/W															
初始值	0x0000															

位	15															0
字段	TRMLCK[15:0]															
属性	R/W															
初始值	0x0001															

寄存器功能

[bit31:0] TRMLCK[31:0]：寄存器写入保护位

bit31:0	描述
读取时	读取 0x00000000 时，MCR_FTRM/MCR_TTRM 寄存器当前未锁定。 读取 0x00000001 时，MCR_FTRM/MCR_TTRM 寄存器当前被锁定。
写入 0x1ACCE554 之外的值	锁定 MCR_FTRM/MCR_TTRM 寄存器
写入 0x1ACCE554	解锁 MCR_FTRM/MCR_TTRM 寄存器

注意事项：

- *软件复位不能初始化本寄存器。*

6. 高速 CR 调节功能使用注意事项

本节说明使用高速 CR 调节功能的注意事项。

- 低速 CR 振荡器

  只有高速 CR 振荡器才能使能此调节功能。
  此调节功能不适用于低速 CR 振荡器。

- "CR 调节" 区中储存的数据

  "CR 调节" 区储存出厂预设频率调节数据。 有关 "CR 调节" 区的地址，参见所用产品的《闪存编程手册》。
  擦除闪存中的数据时，同时也擦除 "CR 调节" 区中的数据。 要使用 "CR 调节" 区中的数据，在擦除闪存中的数据之前，将 "CR 调节" 区中的数据储存到其他区中，比如 RAM 。
  否则，擦除 "CR 调节" 区之外的扇区。

- 高速 CR 振荡器的振荡频率精确度

  如果不设置 CR 振荡温度调节寄存器 (MCR_TTRM) 和高速 CR 振荡频率调节寄存器 (MCR_FTRM)，无法保证《数据手册》所述的高速 CR 振荡器精确度。 因此，使用之前务必设置上述寄存器。

- 如何使用基本计时器

  有关如何使用基本计时器，参见 "计时器部分" 中的 "基本计时器" 和 "I/O 端口" 章节。

- FCS（异常频率检测）

  有关 FCS 功能（异常频率检测），参见 "时钟监视器" 一章。 使能 FCS 功能之后，不得执行 CR 调节。

第 2-4 章：低速 CR 预分频器

本章说明低速 **CR** 预分频器的功能和操作。

1. 低速 **CR** 预分频器概述
2. 低速 **CR** 预分频器配置
3. 低速 **CR** 预分频器操作及设置步骤示例
4. 低速 **CR** 预分频寄存器

代码：9BFLCPC-FM0-C03.0

1. 低速 CR 预分频器概述

本节概述低速 CR 预分频器

低速 CR 预分频器

通过设置低速 CR 预分频器装载寄存器 (LCR_PRSLD)，将低速 CR 预分频为低速 CR 并生成低速 CR 时钟 (CLKLC)。

本模块可以校正低速 CR 的精确度。 有关校正方法，参见校正低速 CR 示例。

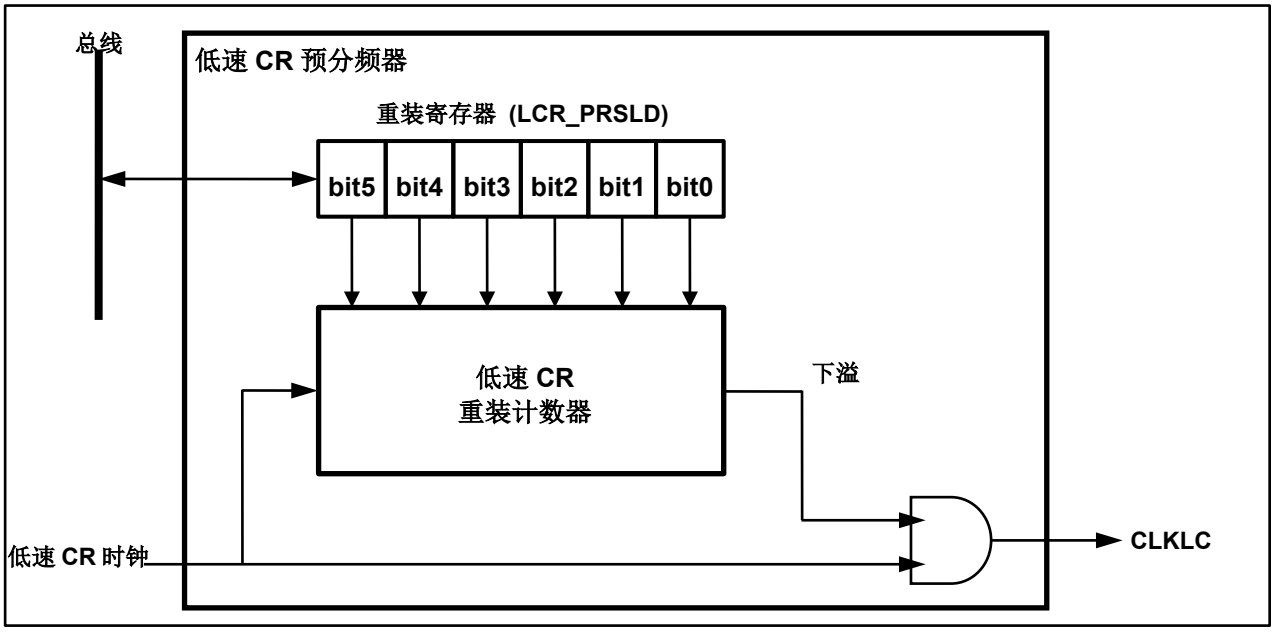
2. 低速 CR 预分频器配置

本节所示为低速 CR 预分频器框图。

低速 CR 预分频器框图

有关低速 CR 预分频器的框图，参见 Figure 2-1。

Figure 2-1 低速 CR 预分频器框图



- 低速 CR 预分频器装载寄存器 (LCR_PRSLD)
  设置低速 CR 预分频器的分频比（重装值）。
- 低速 CR 重装计数器
  这是递减计数器，用于生成低速 CR 分频时钟 (CLKLC)。

3. 低速 CR 预分频器操作及设置步骤示例

本节说明低速 CR 预分频器的操作。 本节同时说明设置步骤示例。

3.1 低速 CR 预分频器设置步骤

低速 CR 与外设时钟 (PCLK) 异步。

要写入低速 CR 预分频器重装寄存器，使用外设时钟。 因此，如果在改变低速 CR 预分频器装载寄存器设置的同时对重装计数器进行重新装载，无法保证重装计数器的重装值。

因此，重写低速 CR 预分频器重装寄存器时，应遵循以下步骤。

■ 切换分频时钟时

低速 CR 预分频器重装寄存器 (LCR-PRSLD) 的初始值为 “0”。

因此，改变初始值时，不需要执行这些步骤。

1. 将低速 CR 预分频器重装寄存器 (LCR_PRSLD) 设置为 "0"。
2. 等待到低速 CR 预分频器重装寄存器 (LCR_PRSLD) 的值被重新载入重装计数器。
3. 使用下述方程式计算等待时间：
   低速 CR 周期 (50 kHz: 20 µs) × "在项目 1 中变更为 "0" 之前的设置值。"
4. 将新的设置值写入低速 CR 预分频器重装寄存器 (LCR_PRSLD)。

有关设置变更时的等待时间，参见 Table 3-1。

Table 3-1 设置等待时间

设置前的重装值	设置值	等待时间
0	0	不存在。
1	0	20 µs (20 µs × 1)
2	0	40 µs (20 µs × 2)
3	0	60 µs (20 µs × 3)
:	:	:
60	0	1200 µs (20 µs × 60)
61	0	1220 µs (20 µs × 61)
62	0	1240 µs (20 µs × 62)
63	0	1260 µs (20 µs × 63)

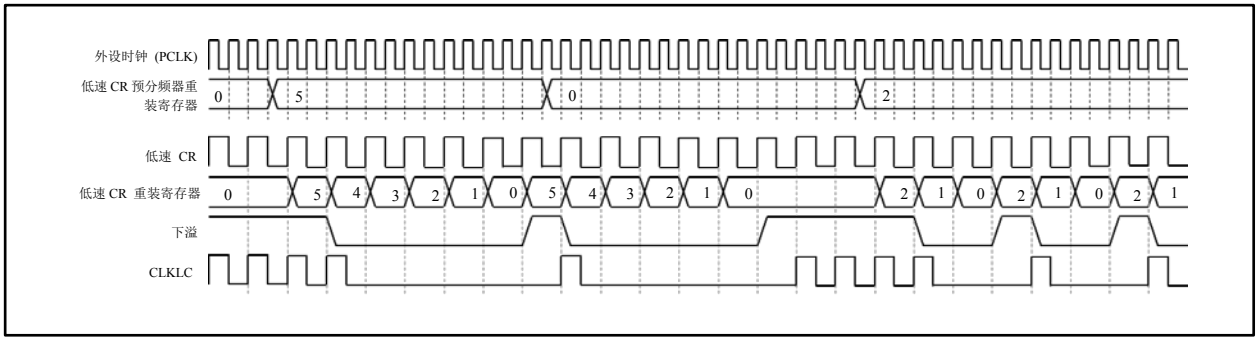
注意事项：

- *分频时钟不能停止。*
- *低速 CR 预分频器重装寄存器 (LCR_PRSLD)在低速 CR 重装计数器下溢时设置。*

3.2 低速 CR 预分频器操作

有关低速 CR 预分频器操作，参见 Figure 3-1。

Figure 3-1 低速 CR 预分频器操作

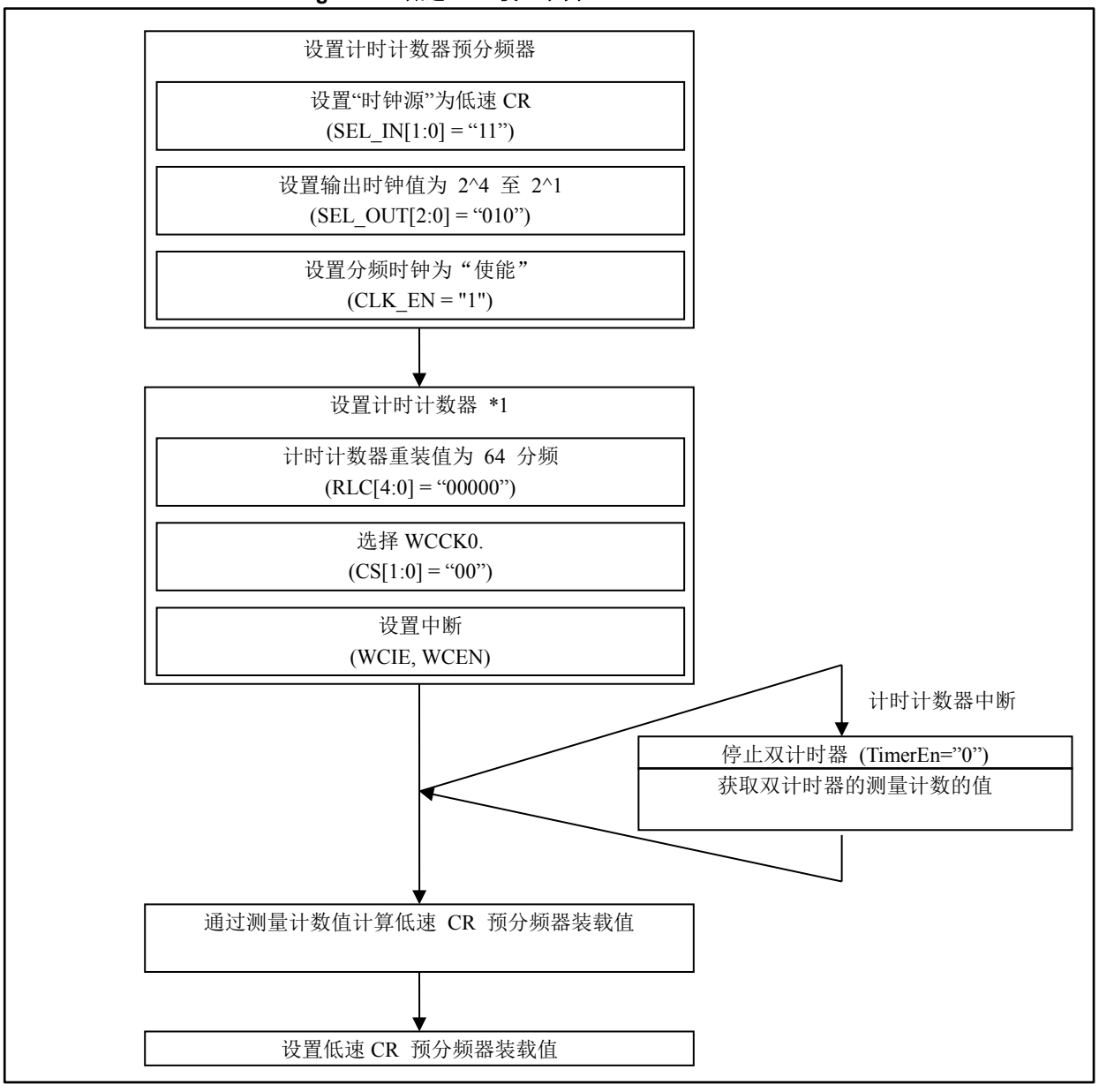


(1) 同步设置低速 CR 预分频器装载寄存器 (LCR_PRSLD) 和外设时钟 (PCLK)。
(2) 在低速 CR 重装计数器指示 "0" 时，检索低速 CR 预分频器装载寄存器 (LCR_PRSLD) 的值。
(3) 在低速 CR 重装计数器下溢时，输出低速 CR (CLKLC)。

3.3 低速 CR 校正示例

有关低速 CR 校正示例，参见 Figure 3-2。

Figure 3-2 低速 CR 校正示例



*1：以上为使用双计时器时的示例。 可使用基本计时器或 MFT 进行测量。

4. 低速 CR 预分频寄存器

本节所示为低速 CR 预分频寄存器列表。

低速 CR 预分频寄存器

Table 4-1 低速 CR 预分频寄存器

缩写	寄存器名称	参考章节
LCR_PRSLD	低速 CR 预分频器控制寄存器	4.1

4.1 低速 CR 预分频器控制寄存器 (LCR_PRSLD)

低速 CR 预分频器控制寄存器用于设置低速 CR 的分频比。

位	7	6	5	4	3	2	1	0
字段	保留		LCR_PRSLD[5:0]					
属性	-		R/W					
初始值	00		000000					

[bit7:6] 保留：保留位

始终读出 "0"。

写入模式时无效。

[bit5:0] LCR_PRSLD：低速 CR 预分频器装载

写入时，设置低速 CR 预分频器（重装计数器的重装值）的分频比。

读取时，读取设置值。

注意事项：

- *软件复位不能初始化本寄存器。*

第 3 章：时钟监视器

本章说明时钟监视器功能。

1. 概述
2. 配置及框图
3. 操作说明
4. 设置步骤示例
5. 操作示例
6. 寄存器
7. 使用注意事项

代码：9BFCSV-FM0-C03.0

1. 概述

本节概述时钟监视器功能。

时钟监视器包括以下两类功能。

时钟故障检测（CSV：时钟监视器的时钟故障检测）

时钟故障检测功能用于监视主时钟和副时钟。 如果在指定周期内未检测到被监视时钟的上升沿，则本功能判定振荡器失效并输出系统复位请求。

异常频率检测（FCS：时钟监视器的异常频率检测）

异常频率检测功能用于监视主时钟的频率。 在高速 CR 分频时钟一边沿和下一边沿之间的指定周期内，本功能通过主时钟计数内部计数器值。 计数值超出设置窗口范围时，本功能判定主时钟出现异常频率并输出中断请求至 CPU 或输出系统复位请求。

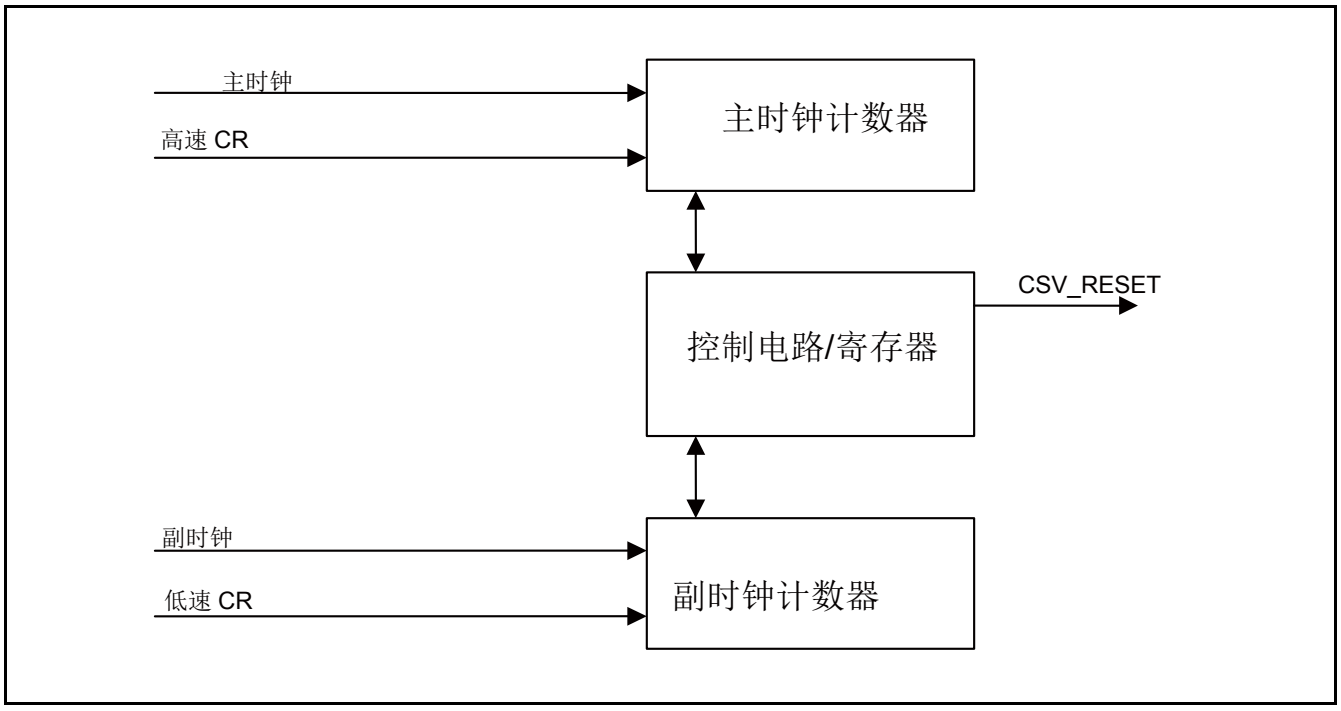
2. 配置及框图

本节说明时钟监视器功能框图。

2.1 时钟故障检测

Figure 2-1 所示为时钟故障检测框图。

Figure 2-1 时钟故障检测框图



时钟故障检测包含以下三类模块。

控制电路/寄存器

- 本模块包括控制时钟故障检测的电路，
- 还包括使能/禁用时钟故障检测的设置寄存器。

主时钟计数器

本计数器通过高速 CR 时钟监视主时钟。

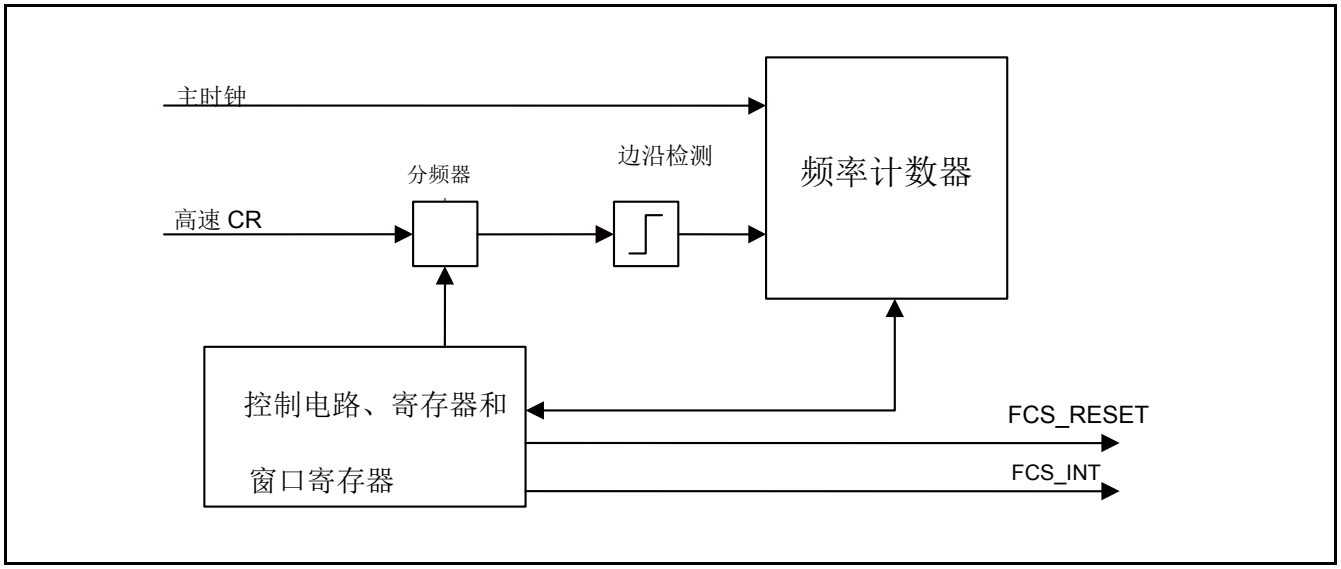
副时钟计数器

本计数器通过低速 CR 时钟监视副时钟。

2.2 异常频率检测

Figure 2-2 所示为异常频率检测框图。

Figure 2-2 异常频率检测框图



异常频率检测包含以下三类模块。

控制电路/寄存器和窗口寄存器

- 本模块包括控制异常频率检测的电路。
- 还包括使能/禁用异常频率检测的设置寄存器。
- 还包括定义测量频率范围的窗口寄存器。

频率计数器

基于主时钟的计数器。

分频器/边沿检测

- 本单元对高速 CR 进行分频。
- 同时检测高速 CR 分频时钟的上升沿。

3. 操作说明

本节说明时钟监视器功能的操作。

时钟故障检测功能

时钟故障检测功能用于监视主时钟和副时钟。 如果在指定周期内未检测到被监视时钟的上升沿，则本功能判定振荡器失效并输出系统复位请求。

- 本复位请求称为 CSV 复位请求。
- CSV 功能独立监视各主时钟和副时钟。
- 主振荡器和副振荡器停止振荡时，停止监视。
- 在等待振荡稳定的等待时间时，将停止监视。
- 等待主振荡器和副振荡器振荡稳定的等待时间结束时，CSV 功能将自动使能。

注意事项：

- *可使用 CSV 控制寄存器 (CSV_CTL) 独立使能/禁用各主时钟故障检测功能和各副时钟故障检测功能。*
- *通过高速 CR 时钟监视主时钟，通过低速 CR 时钟监视副时钟。 在主时钟高速 CR 的 32 个时钟期内，或在副时钟低速 CR 的 32 个时钟期内未检测到上升沿时，本功能判定振荡器失效。*

异常频率检测功能

异常频率检测功能用于监视主时钟。

在高速 CR 分频时钟一上升沿和下一上升沿之间的指定周期内，本功能使用主时钟进行内部计数器计数。计数值超出设置窗口范围时，本功能判定主时钟出现异常频率并输出中断请求或系统复位请求至 CPU 。

- 本中断请求称为 FCS 中断请求，复位请求称为 FCS 复位请求。
- FCS 功能仅用于监视主时钟的频率。
- 主振荡器停止振荡时，停止监视。
- 在等待振荡稳定的等待时间，将停止监视。
- FCS 功能通过一种用户程序软件。

注意事项：

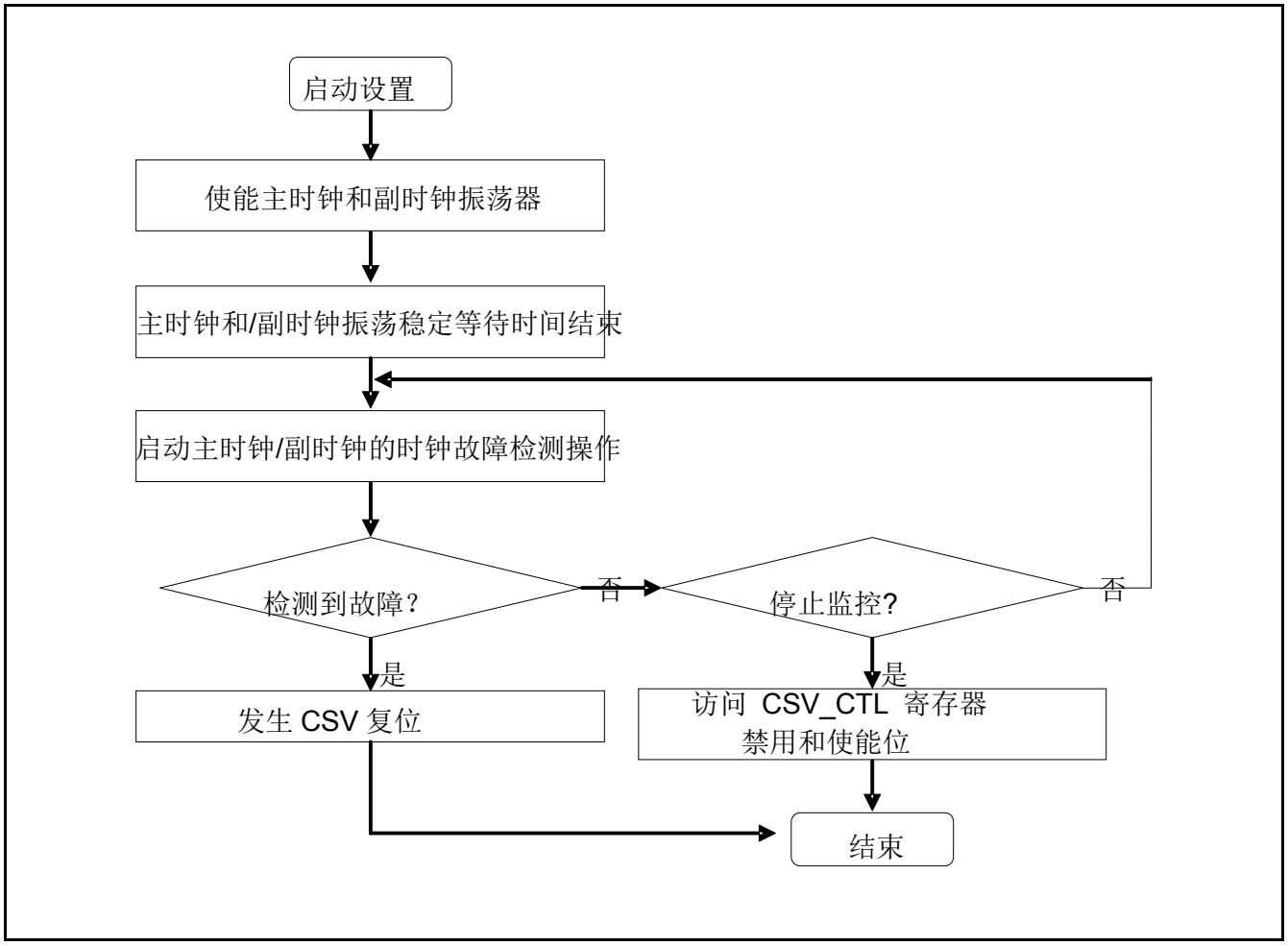
- *如果使能 FCS 复位：*

  计数器值第一次偏离设置窗口时将发生中断请求。 如果中断请求还未清除，且计数器值超出指定窗口，将输出系统复位请求。
  如果未使能 FCS 复位，复位请求被屏蔽。
- *计数器值如果超出指定窗口，则被储存在频率检测计数器寄存器 (FCSWD_CTL) 中。*

4. 设置步骤示例

本节说明时钟监视器功能设置示例。

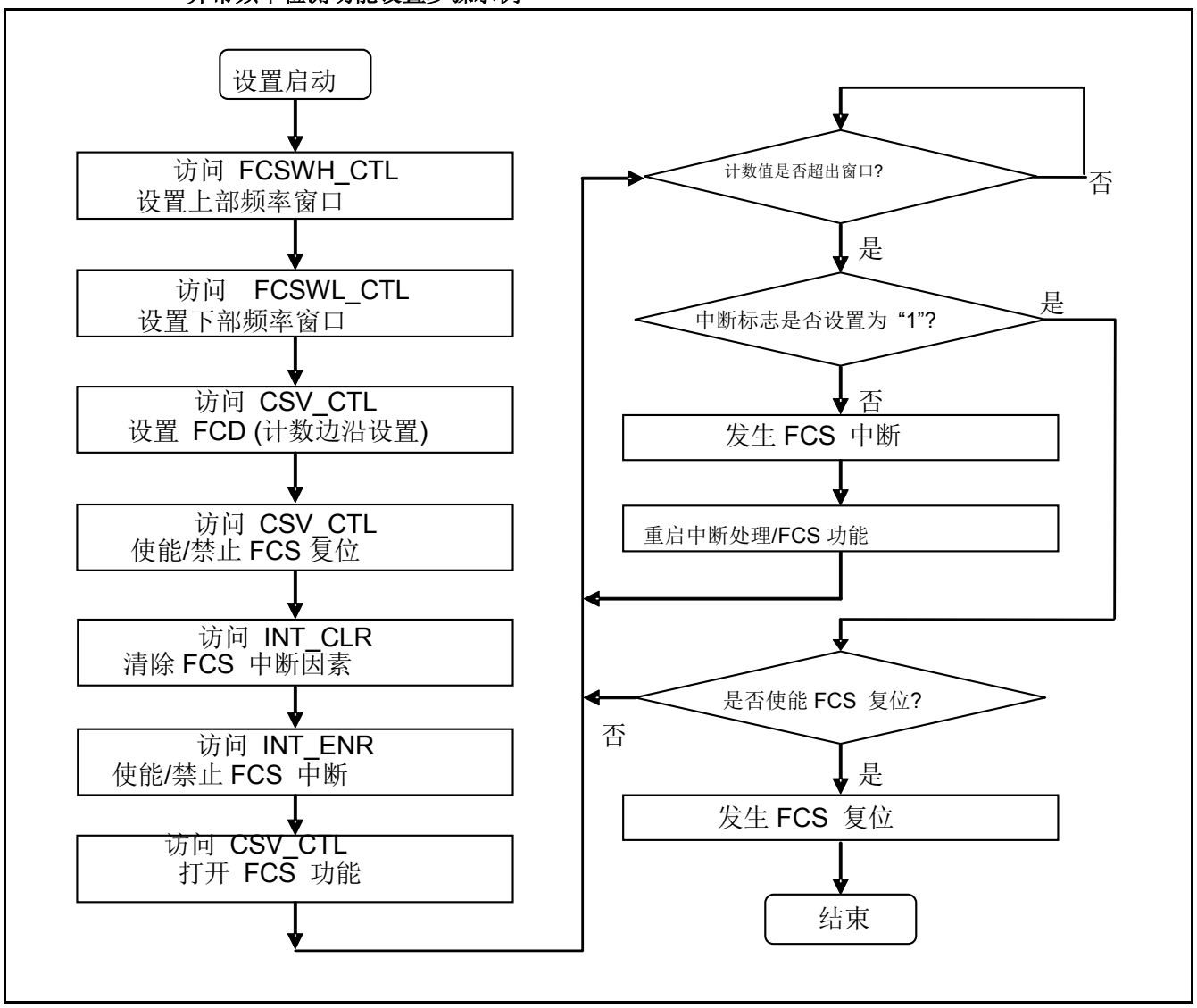
时钟故障检测功能设置步骤示例



注意事项：

- *VBAT 寄存器副振荡控制寄存器 (WTOSCCNT.SOSCNTL) 的 32 kHz 振荡时钟控制链接位从 1 变为 0 时，在副时钟振荡稳定等待结束后，写入寄存器值。*
- *关闭 CHIP 侧的电源仅操作 VBAT 域时，设置 WTOSCCNT.SOSCNTL=0，然后关闭 CHIP 侧的电源。 此外，关闭 CHIP 侧的电源之后，副时钟监视器功能不工作。*
- *有关 VBAT 的详细说明，参见 VBAT 域一章。*

异常频率检测功能设置步骤示例



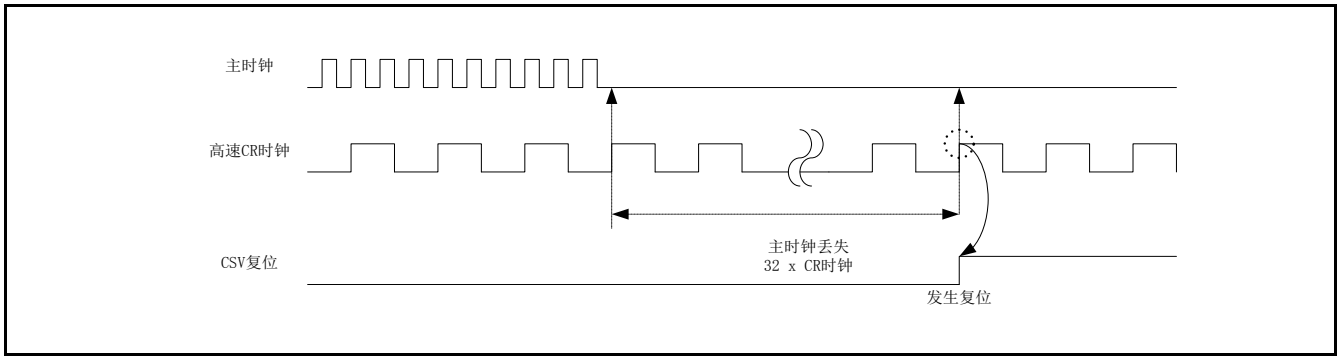
5. 操作示例

本节说明时钟监视器操作示例。

5.1 时钟故障检测

Figure 5-1 所示为时钟故障检测操作示例。

Figure 5-1 时钟故障检测操作示例



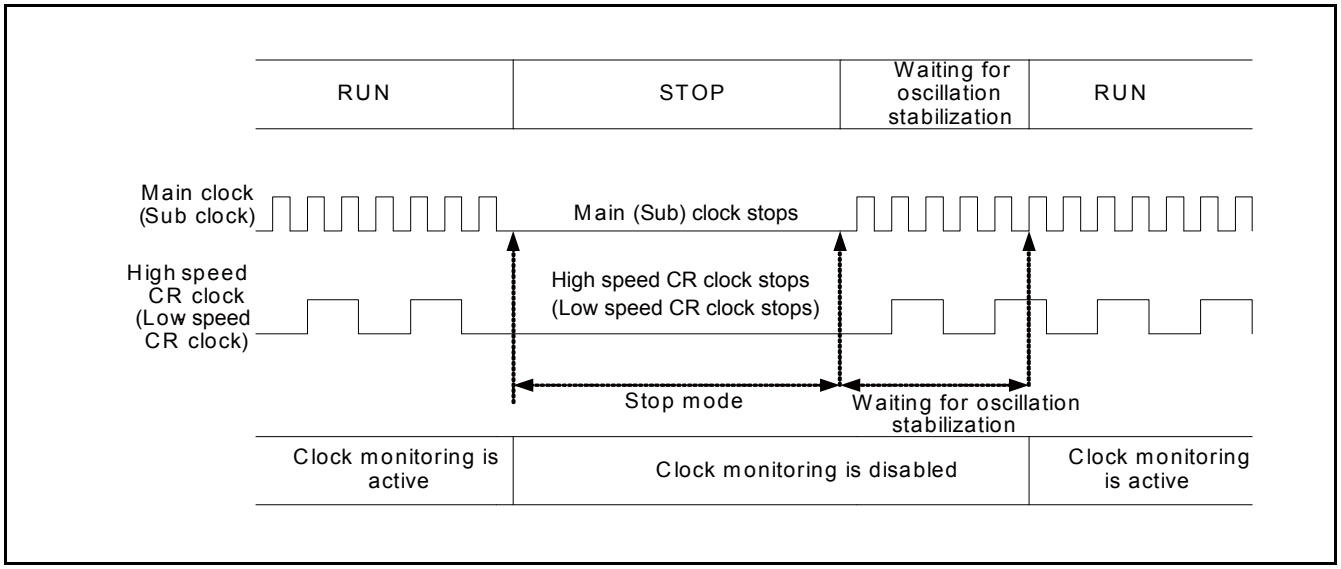
1. 主时钟因故障停止。
2. 本功能通过高速 CR 时钟进行时钟计数。
3. 如果主时钟在高速 CR 的 32 个时钟期内一直停止，本功能判定时钟失效并发出 CSV 复位指令。

注意事项：

- *使用副时钟监视时，如果副时钟在低速 CR 的 32 个时钟期内一直停止，本功能判定该副时钟失效。*

Figure 5-2 所示为停止模式下的时钟故障检测操作示例。

Figure 5-2 停止模式下的时钟故障检测操作示例



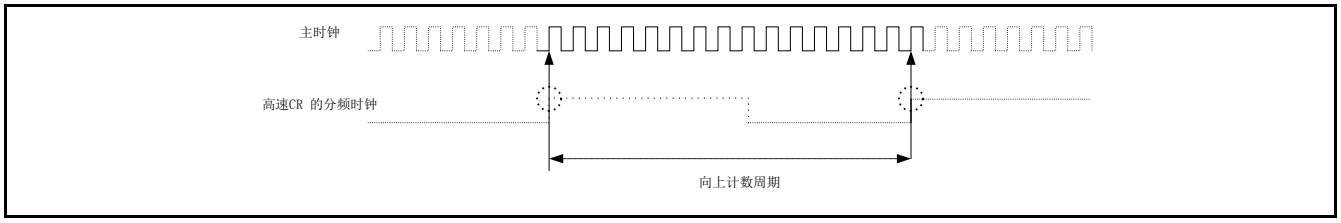
1. 在停止模式下，主时钟和高速 CR 时钟停止。
   同时，时钟监视功能也停止。
2. 释放停止模式后，将立即重启主时钟和高速 CR 时钟振荡，并等待振荡稳定。 同时，时钟监视功能保持停止。

3. 等待振荡稳定的等待时间结束时，将重启时钟监视。

5.2 异常频率检测

Figure 5-3 所示为异常频率检测功能操作示例。

Figure 5-3 异常频率检测功能操作示例



1. 本功能用于检测高速 CR 分频时钟的上升沿。
2. 边沿检测后，使用主时钟进行时钟计数。
3. 在检测高速 CR 分频时钟的下一上升沿之前，保持计数。
4. 使得 "α"为主时钟时的计数值。
   同时使得 A 表示下限窗口值， B 表示上限窗口值。 将计数值□与这些窗口值比较，如果方程式 A≤α ≤ B
   成立，则可判定频率正常。
   如果计数值 □超出范围，即
   □α<A 或 B<α□
   则可判定频率异常并中断。
   如果在中断之后还未清除中断标志并且再次检测到异常频率，本功能将根据设置发出复位指令。

5.3 异常频率检测功能窗口设置示例

异常频率检测在高速 CR 分频时钟沿之间计数。 CR 精确度将影响测量间隔。 配置窗口寄存器值时，必须考虑 CR 精确度。
有关 CR 振荡器的频率精确度，参见相关 《数据手册》。

计算方法

异常频率检测计数值范围必须增加 CR 精确度，然后设置窗口寄存器值。必须使用下述计数范围方程式。

$$\text{计数值} = \left(\frac{1}{\text{CR的分频时钟频率} \times \left(\frac{1}{1 \pm \frac{\text{CR 精度}}{100}} \right)} \right) \times \text{主时钟频率}$$

通过频率 L [Hz] 的主时钟进行的计数值可使用 Y 分频的精确度为 ±Z% 频率 K [Hz]的 CR 振荡器时钟计算。

计数值 A（CR 频率正精确度）= 1/[(K/Y) ×(1 + Z/100)] × L
计数值 B（CR 频率负精确度）= 1/[(K/Y) × (1 - Z/100)] × L

在这些方程式计算 A~ B 范围内增加内部 CR 精确度的计数值。
窗口下限设置为小于计数值 A 的值，上限设置为大于计数值 B 的值。

窗口设置通过用户定义的主振荡频率波动允许值确定。

示例计算

通过频率 4MHz 的主时钟进行的计数值可使用精确度为 ±5% 且频率为 4MHz 的 1024 分频 CR 振荡器时钟计算计数值。

计数值 A（CR 频率正精确度）

$$\text{计数值 A} = \left(\frac{1}{\frac{4\times10^6}{1024} \times \left(1+ \frac{5}{100} \right)} \right) \times 4\times10^6 \approx 975$$

计数值 B（CR 频率负精确度）

$$\text{计数值 B} = \left(\frac{1}{\frac{4\times10^6}{1024} \times \left(1- \frac{5}{100} \right)} \right) \times 4\times10^6 \approx 1078$$

这些方程式计算出的计数值在 975 ~ 1078 的范围内，包括高速 CR 误差。 如果窗口设置值为 5%，则窗口设置值如下。

窗口下限 = 975×0.95(-5%) = 926.25≈3.43MHz
窗口上限 = 1078×1.05(+5%) = 1131.9≈4.64MHz

因此，可以判定超出 3.4MHz ~ 4.6MHz 范围的主时钟频率异常。Table 5-1 列出窗口设置示例。

Table 5-1 窗口设置示例

高速 CR 分频时钟	主时钟	高速 CR 误差	计数值(包括高速 CR 误差)	窗口设置值下限	窗口设置值上限
CR 的 1024 分频时钟：4MHz	4MHz	±5%	975 (≈3.61MHz) - 1078 (≈4.42MHz)	926 (≈3.43MHz)	1131 (≈4.64MHz)

6. 寄存器

本节说明时钟监视器功能寄存器表。

寄存器表

Table 6-1 列出寄存器表。

Table 6-1 寄存器表

缩写	寄存器名称	参考章节
CSV_CTL	CSV 控制寄存器	6.1
CSV_STR	CSV 状态寄存器	6.2
FCSWH_CTL	频率检测窗口设置寄存器（上限）	6.3
FCSWL_CTL	频率检测窗口设置寄存器（下限）	6.4
FCSWD_CTL	频率检测计数器寄存器	6.5

6.1 CSV 控制寄存器 (CSV_CTL)

The CSV_CTL 寄存器配置 CSV 功能的控制。

寄存器配置

位	15	14	13	12	11	10	9	8
字段	保留		FCD		保留		FCSRE	FCSDE
属性	-		R/W		-		R/W	R/W
初始值	0		111		00		0	0

位	7	6	5	4	3	2	1	0
字段	保留						SCSVE	MCSVE
属性			-				R/W	R/W
初始值			000000				1	1

寄存器功能

[bit15] 保留：保留位

此位读取值为 0.

写入时将此位设置为 0.

[bit14:12] FCD：FCS 计数周期设置位

bit14:12	描述
写入 000 时	禁止设置
写入 001 时	
写入 010 时	
写入 011 时	
写入 100 时	
写入 101 时	高速 CR 振荡频率为 1/256
写入 110 时	高速 CR 振荡频率为 1/512
写入 111 时	高速 CR 振荡频率为 1/1024[初始值]
读取时	读取寄存器值。

[bit11:10] 保留：保留位

这些位的读取值为 0b00。

写入时将这些位设置为 0b00。

[bit9] FCSRE：FCS 复位输出使能位

位	描述
写入 0 时	禁用 FCS 复位[初始值】
写入 1 时	使能 FCS 复位
读取时	读取寄存器值。

[bit8] FCSDE：FCS 功能使能位

位	描述
写入 0 时	禁用 FCS 功能□初始值□
写入 1 时	使能 FCS 功能
读取时	读取寄存器值。

[bit7:2] 保留：保留位

这些位的读取值为 0b000000。

写入时把这些位设置为 0b000000。

[bit1] SCSVE：副 CSV 功能使能位

位	描述
写入 0 时	禁用副 CSV 功能
写入 1 时	使能副 CSV 功能 [初始值]
读取时	读取寄存器值。

[bit0] MCSVE：主 CSV 功能使能位

位	描述
写入 0 时	禁用主 CSV 功能
写入 1 时	使能主 CSV 功能 [初始值]
读取时	读取寄存器值。

注意事项：

- *软件复位不能初始化本寄存器。*
- *使能副时钟监视器功能时，将系统时钟模式控制寄存器 (SCM_CTL.SOSCE) 的副时钟振荡使能设置为“1”，等待系统时钟模式控制寄存器 (SCM_STR.SORDY) 的副时钟振荡稳定位变为“1”。*
- *仅使能 VBAT RTC 副时钟振荡时 (WTOSCCNT.SOSCEX=0) ，副时钟监视器功能不工作。*
- *禁止以下设置组合：*
  VBAT RTC 副时钟振荡控制寄存器 (WTOSCCNT.SOSCNTL) 的 32 kHz 振荡时钟控制链接位为“0”。
  VBAT RTC 副时钟振荡控制寄存器 (WTOSCCNT.SOSCEX) 的 32 kHz 振荡使能位为“1”。
  系统时钟模式控制寄存器 (SCM_CTL.SOSCE) 的副时钟振荡使能设置位为“1”。
  CSV 控制寄存器 (CSV_CTL.SCSVE) 的副 CSV 功能使能位 = “1”。
- *有关 VBAT RTC 的详细说明，参见 "VBAT 域" 一章。*

6.2 CSV 状态寄存器 (CSV_STR)

CSV_STR 寄存器指示 CSV 功能状态。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	保留						SCMF	MCMF
属性	-						R	R
初始值	000000						0	0

寄存器功能

[bit7:2] 保留：保留位

这些位的读取值为 0b000000。

写入时把这些位设置为 0b000000。

[bit1] SCMF：副时钟故障检测位

位	描述
写入时	无效
读出 "0" 时	未检测到任何副时钟故障。 [初始值]
读出 1 时	检测到副时钟故障。

[bit0] MCMF：主时钟故障检测标志

位	描述
写入时	无效
读出 0 时	未检测到任何主时钟故障。 [初始值]
读出 1 时	检测到主时钟故障。

注意事项：

- *读取时将清除寄存器。*

6.3 频率检测窗口设置寄存器（上限）(FCSWH_CTL)

FCSWH_CTL 寄存器配置频率检测窗口设置寄存器（上限）。

寄存器配置

位	15															0
字段	FWH															
属性	R/W															
初始值	0xFFFF															

寄存器功能

[bit15:0] FWH：频率检测窗口设置位（上）

bit15:0	描述
写入时	这些位可以写入任何值。
读取时	读取寄存器值。

注意事项：

- *设置的值大于 FCSWL_CTL（频率检测窗口设置寄存器（下限））所设值。*
- *软件复位不能初始化本寄存器。*

6.4 频率检测窗口设置寄存器（下限）(FCSWL_CTL)

FCSWL_CTL 寄存器配置频率检测窗口设置寄存器。

寄存器配置

位	15															0
字段	FWL															
属性	R/W															
初始值	0x0000															

寄存器功能

[bit15:0] FWL：频率检测窗口设置位（下限）

bit15:0	描述
写入时	这些位可以写入任何值。
读取时	读取寄存器值。

注意事项：

- *设置值小于 FCSWH_CTL（频率检测窗口设置寄存器（上限））所设值。*
- *软件复位不能初始化本寄存器。*

6.5 频率检测计数器寄存器 (FCSWD_CTL)

FCSWD_CTL 寄存器通过主时钟指示频率检测计数值。

寄存器配置

位	15															0
字段	FWD															
属性	R															
初始值	0x0000															

寄存器功能

[bit15:0] FWD：频率检测计数数据

bit15:0	描述
写入时	操作无效
读取时	读取计数值。

注意事项：

- *检测到误差时，本寄存器将保留计数值。*
- *软件复位不能初始化本寄存器。*

7. 使用注意事项

本节说明使用时钟监视器功能的注意事项。

- 有关使能和清除频率检测中断源的详细说明，参见 "时钟" 一章。

- 有关时钟故障检测和异常频率检测复位源的详细说明，参见 "复位" 一章。

- 复位后的操作

  时钟故障检测触发复位后，时钟模式返回高速 CR。
  不得再次选择故障时钟。

- 频率检测高速 CR 时钟

  频率故障检测受高速 CR 本身的频率精确度影响。
  因此，配置频率窗口时，必须考虑窗口值相关的高速 CR 精确度。使能异常频率检测后不得调节高速 CR 时钟。

- 使用前的异常频率检测设置顺序

  使能 FCS (FCSDE=1) 之前，指定计数周期 (FCD)、使能复位 (FCSRE) 、设置频率窗口 (FWH/FWL)。
  如果要在使能 FCS 后改变 FCD/FCSRE/FWH/FWL，改变设置前停止 FCS 功能。FCS 使能时，不得改变设置。

- 使用前的异常频率检测使能设置

  在异常频率检测进程中，将根据 CSV 控制寄存器 (CSV_CTL) 中 FCSRE 位的设置改变操作。**Table 7-1** 列出设置表。

Table 7-1 FCS 功能和 FCSRE 位设置表

	FCSRE=0	FCSRE=1
FCSDE=0	停止 FCS 功能	停止 FCS 功能
FCSDE=1	使能 FCS 功能 根据误差检测生成中断	使能 FCS 功能 根据第一次误差检测生成中断 根据第二次误差检测生成复位

- 频率检测中断设置和主计时器模式

  时钟模式为主计时器模式时，内部总线时钟停止。 在本模式下，即使在 FCSRE 被设置为 0 时检测到误差，也不会发生中断。
  因此，在主计时器模式下，不得将 FCSRE 位设置为 "0"。 如果 FCSRE 位被设置为 1，根据第二次误差检查生成复位。

- CSV OFF 和外部复位设置。

  CSV 功能被设置为 OFF 时，即使时钟出现故障，不会生成 CSV 复位，也不会生成外部复位 (INITX) 。 因此，若无特殊原因，建议不关闭 CSV 功能。

第 4 章：复位

本章说明复位功能及操作。

1. 概述
2. 配置
3. 操作说明
4. 寄存器

代码：9AFRESET-FM0-C03.0

1. 概述

本系列产品具有以下复位因素。接收到复位因素后，立即发出复位指令并初始化设备。

- 上电复位
- INITX 引脚输入
- 外部电源/低压检测复位
- 软件看门狗复位
- 硬件看门狗复位
- 时钟故障检测复位
- 异常频率检测复位
- 软件复位
- 深度待机转换复位

本节所示的复位因素不能初始化 VBAT 域。

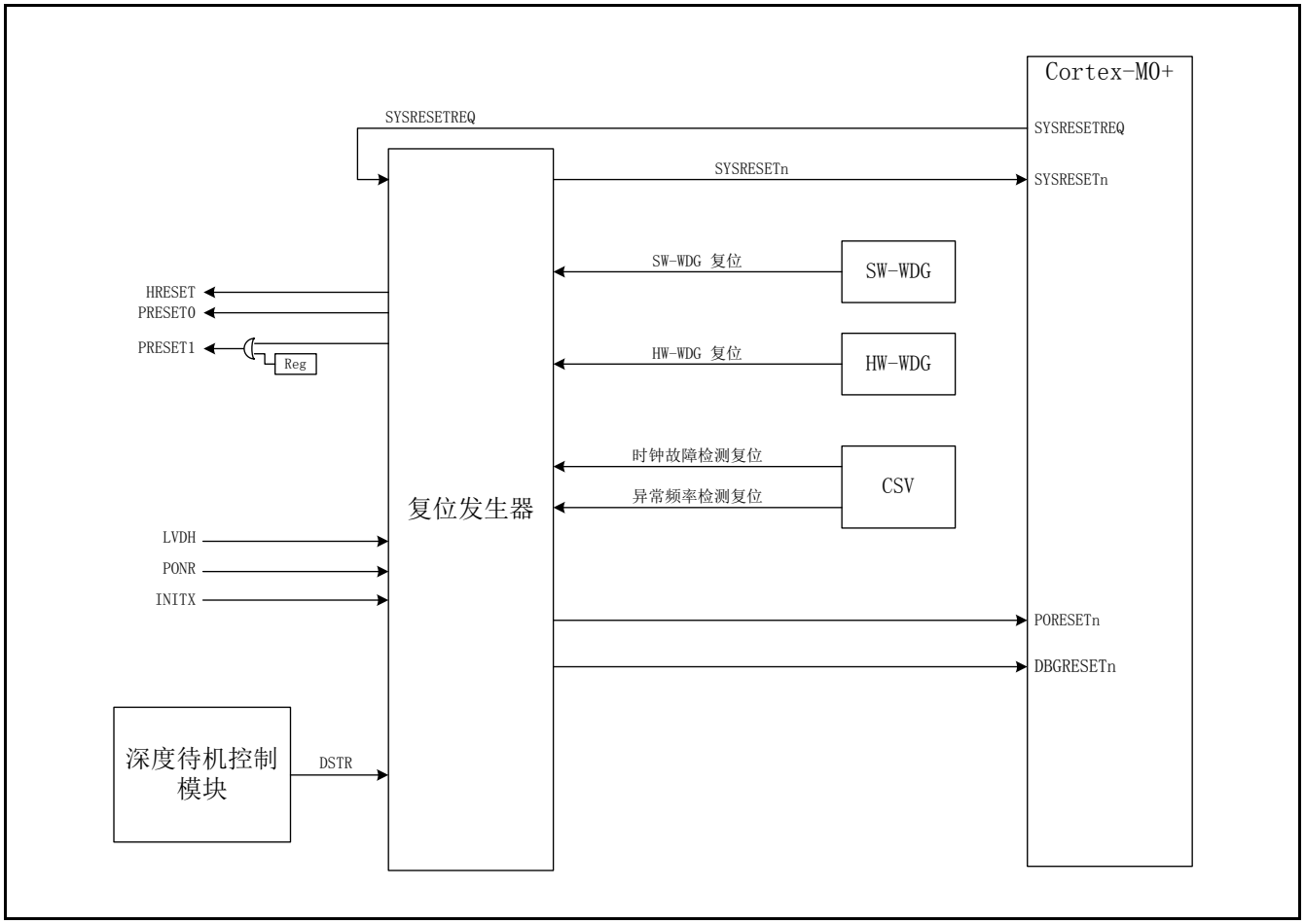
有关 VBAT 域复位，参见 "VBAT 域" 一章。

2. 配置

本节说明复位电路的操作。

复位框图

Figure2-1 复位框图



PONR	：	上电复位
INITX	：	INITX 引脚输入复位
LVDH	：	低压检测复位
HRESET	：	AHB 总线复位（所有复位因素发出的总线复位）
PRESET0, 1	：	APB0、APB1 总线复位（所有复位因素发出的总线复位）
SW-WDG reset	：	软件看门狗复位
HW-WDG reset	：	硬件看门狗复位
CSV 复位	：	时钟故障检测复位
FCS 复位	：	异常频率检测复位
PORESETn	：	输入至 Cortex-M0+ 的上电复位

SYSRESETn	：	输入至 Cortex-M0+ 的系统复位
SYSRESETREQ	：	Cortex-M0+ 内部复位控制寄存器的 "SYSRESETREQ 位" 信号
DBGRESETn	：	SW-DP 复位
DSTR	：	深度待机转换复位

3. 操作说明

本节说明本系列产品的复位操作。

3.1 复位因素
3.2 设备内复位
3.3 复位序列
3.4 清除复位后的操作

3.1 复位因素

本节说明复位因素。

上电复位 (PONR)

上电时生成的复位

生成	通过检测电源的上升沿生成本信号。
清除	发出复位后自动清除本信号。
初始化目标	初始化所有寄存器设置和硬件。
标志	复位因素寄存器 (RST_STR) 的 bit0 (PONR) = 1

INITX 引脚输入复位 (INITX)

设备从外部输入的复位

生成	通过将低电平输入 INITX 引脚生成本信号。
清除	通过将高电平输入 INITX 引脚清除本信号。
初始化目标	初始化所有寄存器设置和硬件，调试电路、深度待机控制模块和 RTC （某些寄存器）除外。 注意事项：不初始化以下寄存器： – 复位因素寄存器 (RST_STR) – 低压检测电压设置寄存器 bit15 至 bit8 – 深度待机唤醒因素寄存器 1 和 2 (WRFSR, WIFSR) – 深度待机 RAM 保留寄存器 (DSRAMR) – 备用寄存器 01 至 16（BUR01 至 16）
标志	复位因素寄存器 (RST_STR) 的 bit1 (INITX) = 1

* 如果复位是从 INITX 引脚的异步输入，保留片上 SRAM 的内容。

低压检测复位、外部电压监控 (LVDH)

检测到外部电压降低时，从低压检测电路输入的复位。

生成	当外部电压低于指定电平时，产生此信号
清除	当外部电压高于指定电平时，清除此信号
初始化目标	初始化所有寄存器设置和硬件。
标志	复位因素寄存器 (RST_STR) 的 bit0 (PONR) = 1

软件看门狗复位 (SWDGR)

从软件看门狗计时器输入的复位。

生成	软件看门狗计时器下溢时将生成本信号。
清除	发出复位后自动清除本信号。
初始化目标	初始化所有寄存器设置和硬件，调试电路和硬件看门狗计时器（包括控制寄存器）和深度待机控制模块除外。 注意事项：不初始化以下寄存器： – 复位因素寄存器 (RST_STR) – 低压检测电压设置寄存器 (LVD_CTL) bit15 至 bit8 – 深度待机唤醒因素寄存器 1 和 2 (WRFSR, WIFSR) – 深度待机 RAM 保留寄存器 (DSRAMR) – 备用寄存器 01 至 16（BUR01 至 16）
标志	复位因素寄存器 (RST_STR) 的 bit4 (SWDT) = 1

硬件看门狗复位 (HWDGR)

从硬件看门狗计时器输入的复位。

生成	硬件看门狗计时器下溢时将生成本信号。
清除	发出复位后自动清除本信号。
初始化目标	初始化所有寄存器设置和硬件，调试电路、深度待机控制模块和 RTC （某些寄存器）除外。 注意事项：不初始化以下寄存器： – 复位因素寄存器 (RST_STR) – 低压检测电压设置寄存器 (LVD_CTL) bit15 至 bit8 – 深度待机唤醒因素寄存器 1 和 2 (WRFSR, WIFSR) – 深度待机 RAM 保留寄存器 (DSRAMR) – 备用寄存器 01 至 16（BUR01 至 16）
标志	复位因素寄存器 (RST_STR) 的 bit5 (HWDT) = 1

时钟故障检测复位 (CSVR)

监测到主晶体振荡器或副晶体振荡器出现故障时输入的复位。

生成	在主晶体振荡器或副晶体振荡器中检测到时钟故障时将生成本信号。
清除	发出复位后自动清除本信号。
初始化目标	初始化所有寄存器设置和硬件，调试电路和时钟故障检测电路（某些寄存器）、深度待机控制模块、和 RTC （某些寄存器）除外。 注意事项：不初始化以下寄存器： – 复位因素寄存器 (RST_STR) – 低压检测电压设置寄存器 (LVD_CTL) bit15 至 bit8 – 深度待机唤醒因素寄存器 1 和 2 (WRFSR, WIFSR) – 深度待机 RAM 保留寄存器 (DSRAMR) – 备用寄存器 01 至 16（BUR01 至 16）
标志	复位因素寄存器 (RST_STR) 的 bit6 (CSVR) = 1 CSV 状态寄存器 (CSV_STR) 的 bit1 (SCMF) 或 bit0 (MCMF) = 1 注意事项： 有关 CSV_STR 的详细说明，参见 “时钟监视器” 一章。

异常频率检测复位 (FCSR)

检测到主晶体振荡器出现异常频率时输入的复位。

生成	主晶体振荡器的频率超出任何指定设置时将生成本信号。
清除	发出复位后自动清除本信号。
初始化目标	初始化所有寄存器设置和硬件，调试电路、异常频率检测（某些寄存器）、深度待机控制模块、和 RTC （某些寄存器）除外。 注意事项：不初始化以下寄存器： – 复位因素寄存器 (RST_STR) – 低压检测电压设置寄存器 (LVD_CTL) bit15 至 bit8 – 深度待机唤醒因素寄存器 1 和 2 (WRFSR, WIFSR) – 深度待机 RAM 保留寄存器 (DSRAMR) – 备用寄存器 01 至 16（BUR01 至 16）
标志	复位因素寄存器 (RST_STR) 的 bit7 (FCSR) = 1

软件复位 (SRST)

访问复位控制寄存器时生成的复位。

生成	写入 Cortex-M0+ 内部复位控制寄存器 (SYSRESETREQ 位) 时将生成本信号。
清除	发出复位后自动清除本信号。
初始化目标	初始化所有寄存器设置和硬件，以下除外： 软件复位不能初始化以下功能和寄存器： – 调试电路 – 深度待机控制模块 – 某些 RTC 寄存器 – 与时钟控制相关的部分寄存器（可初始化外部时钟停止寄存器。） 用来控制软件和硬件看门狗计时器的部分寄存器 – 时钟故障检测电路中的部分寄存器 – 用来检测异常频率的部分寄存器 – 用于 CR 调节的部分寄存器 – 复位因素寄存器 (RST_STR) – 低压检测电压设置寄存器 (LVD_CTL) bit15 至 bit8 – RTC 模式控制寄存器 (PMD_CTL) – 深度待机唤醒因素寄存器 1 和 2 (WRFSR, WIFSR) – 深度待机 RAM 保留寄存器 (DSRAMR) – 备用寄存器 01 至 16（BUR01 至 16）
标志	复位因素寄存器 (RST_STR) 的 bit8 (SRST) = 1

深度待机转换复位 (DSTR)

转换至深度待机模式时出现的复位。

生成	转换至深度待机模式时将生成本信号。
清除	从深度待机模式唤醒时将清除本信号。
初始化目标	初始化所有寄存器设置和硬件，以下除外： 深度待机转换复位不能初始化以下功能和寄存器： – 深度待机控制模块 – 某些 RTC 寄存器 – HDMI-CEC/远控接收器 – 某些 GPIO 寄存器（有关详细说明，参见“I/O 端口”一章） – 低压检测电路寄存器 – RTC 模式控制寄存器 (PMD_CTL) – 深度待机唤醒因素寄存器 1 和 2 (WRFSR, WIFSR) – 深度待机唤醒许可寄存器 (WIER) – WKUP 引脚输入电平寄存器 (WILVR) – 深度待机 RAM 保留寄存器 (DSRAMR) – 备用寄存器 01 至 16（BUR01 至 16）
标志	深度待机唤醒因素寄存器 1 或 2 (WRFSR, WIFSR) 的位被设置为 "1"。 注意事项：变成 "1" 的位随唤醒因素的变化而变化。

注意事项：

- *有关控制软件复位的 Cortex-M0+ 内部复位控制寄存器 (SYSRESETREQ)，参见《ARMv6-M 结构参考手册》中的 "第 B3 章：系统地址映射"。*
- *确定各复位因素发生的复位因素寄存器只能通过上电复位初始化。*

3.2 设备内复位

本节说明本设备的内部复位信号。

从内部与连接至设备的复位被划分为输入至 Cortex-M0+ 内核的复位和输入至外设电路的复位。

3.2.1 复位至 Cortex-M0+
3.2.2 复位至外设电路

3.2.1 复位至 Cortex-M0+

本设备输入至 Cortex-M0+ 的三种复位，分别为 PORESETn、SYSRESETn 和 DBGRESETn。下文说明这三种复位输入的复位因素。

上电复位 PORESETn

复位因素	– 上电复位 (PONR) – 低压检测复位 (LVDH) – 深度待机转换复位 (DSTR)

系统复位 SYSRESETn

复位因素	– 上电复位 (PONR) – 低压检测复位 (LVDH) – INITX 引脚输入 (INITX) – 软件看门狗复位 (SWDGR) – 硬件看门狗复位 (HWDGR) – 时钟故障检测复位 (CSVR) – 异常频率检测复位 (FCSR) – 软件复位 (SRST) – 深度待机转换复位 (DSTR)

SW-DP 复位 DBGRESETn

复位因素	– 上电复位 (PONR) – 低压检测复位 (LVDH) – 深度待机转换复位 (DSTR)

3.2.2 复位至外设电路

基本上所有复位因素都可生成输入至外设电路的总线复位（HRESET、PRESET0 和 PRESET1）。可通过寄存器设置控制 PRESET1 复位。

下文说明总线复位的复位因素。

复位至外设电路

■ HRESET 和 PRESET0

复位因素	– 上电复位 (PONR) – 低压检测复位 (LVDH) – INITX 引脚输入 (INITX) – 软件看门狗复位 (SWDGR) – 硬件看门狗复位 (HWDGR) – 时钟故障检测复位 (CSVR) – 异常频率检测复位 (FCSR) – 软件复位 (SRST) – 深度待机转换复位 (DSTR)

■ PRESET1

复位因素	– 上电复位 (PONR) – 低压检测复位 (LVDH) – INITX 引脚输入 (INITX) – 软件看门狗复位 (SWDGR) – 硬件看门狗复位 (HWDGR) – 时钟故障检测复位 (CSVR) – 异常频率检测复位 (FCSR) – 软件复位 (SRST) – APB 总线复位 (APBC1_PSR) – 深度待机转换复位 (DSTR)

注意事项：

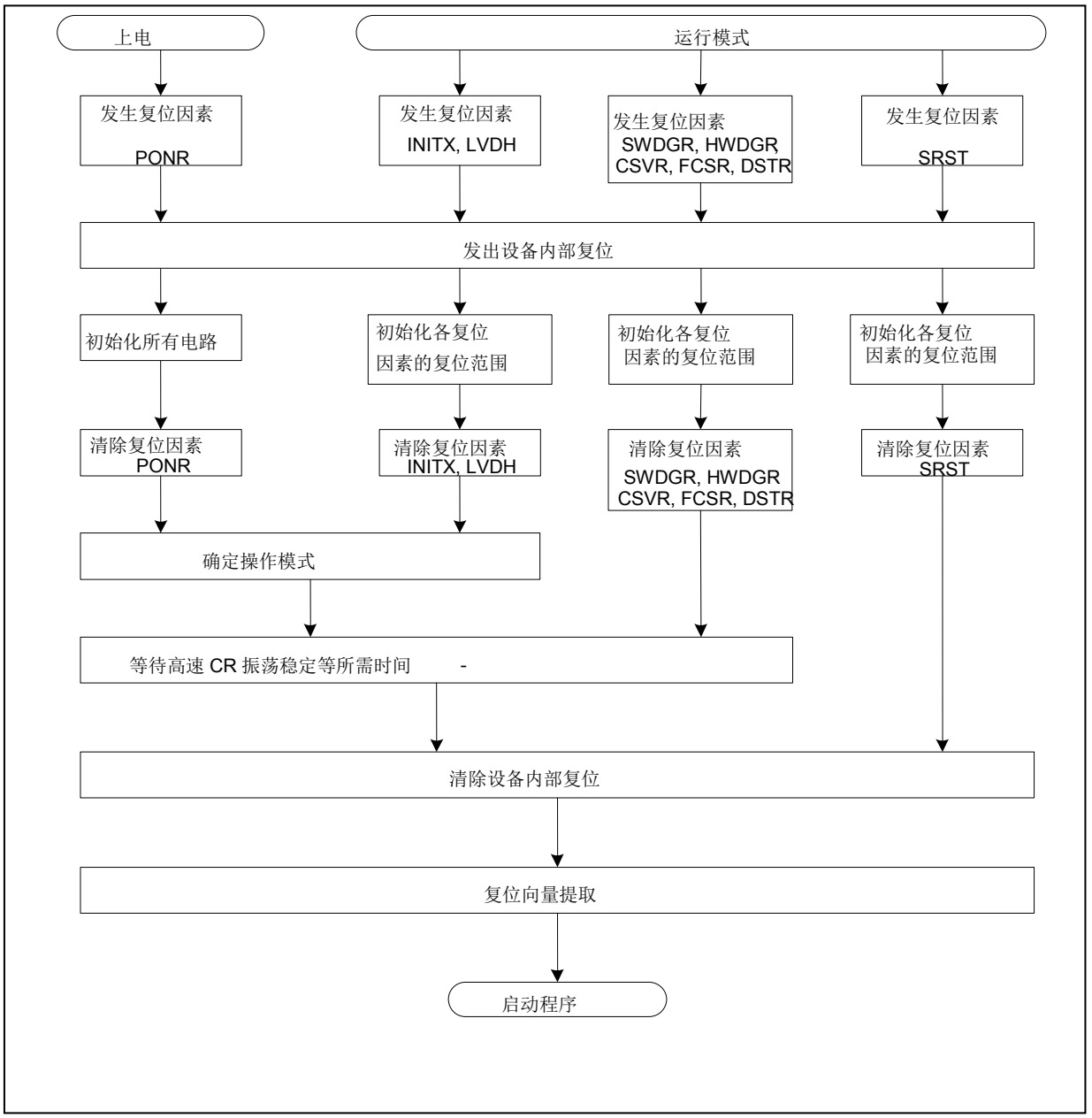
- *本质上可通过所有复位因素初始化外设电路。依据外设电路的规格，存在仅能以指定原因初始化的寄存器。有关寄存器的初始化条件，参见相关章节所述的寄存器初始化条件。*
- *有关 APB 总线复位 (APBC1_PSR) 的详细说明，参见 "时钟" 一章。*

3.3 复位顺序

清除复位因素时，本系列产品将从初始状态开始初始化程序和硬件操作。
从复位开始到初始化操作结束的一些列操作被称为复位顺序。
下文说明复位顺序。

复位状态转换图

下图所示为复位状态转换。有关详细操作，参见下节 3.4 清除复位后的操作。



1. 捕捉复位因素

   捕捉并保留复位因素，直到向设备发出复位指令。

2. 发出复位指令

   已就绪发出复位指令后，发出设备内部复位指令。

3. 清除复位

   清除复位因素时，设备内部复位延续清除复位所需的时间（比如，高速 CR 振荡稳定所需的等待时间）。延续时间结束时，复位被清除。

4. 确定操作模式

   在 PONR、LVDH 或 INITX 清除并通知各硬件后确定 MD0 和 MD1 定义的操作模式。其他任何复位因素不会改变操作模式。

5. 复位向量提取

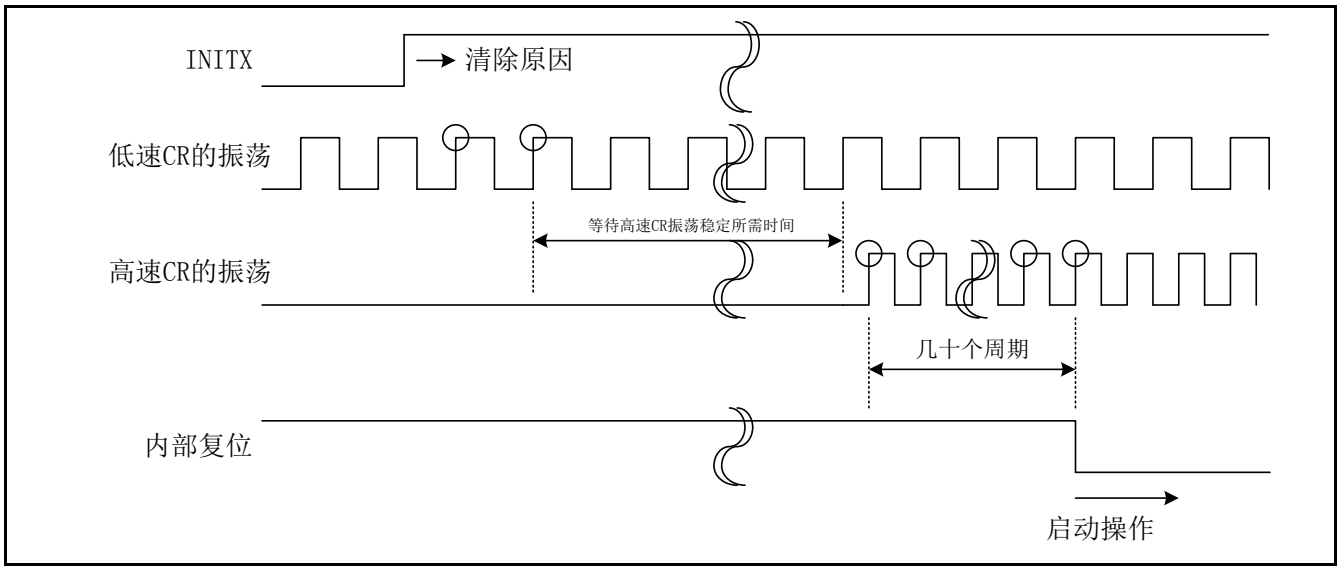
   清除设备内部复位之后，CPU 开始提取复位向量。CPU 将获得的复位向量提取到程序计数器内，开始程控操作。

3.4 清除复位后的操作

PONR, LVDH, INITX, HWDGR, SWDGR, CSVR, FCSR, DSTR

Figure 3-1 所示为 INITX 引脚输入复位被清除后的操作波形示例。

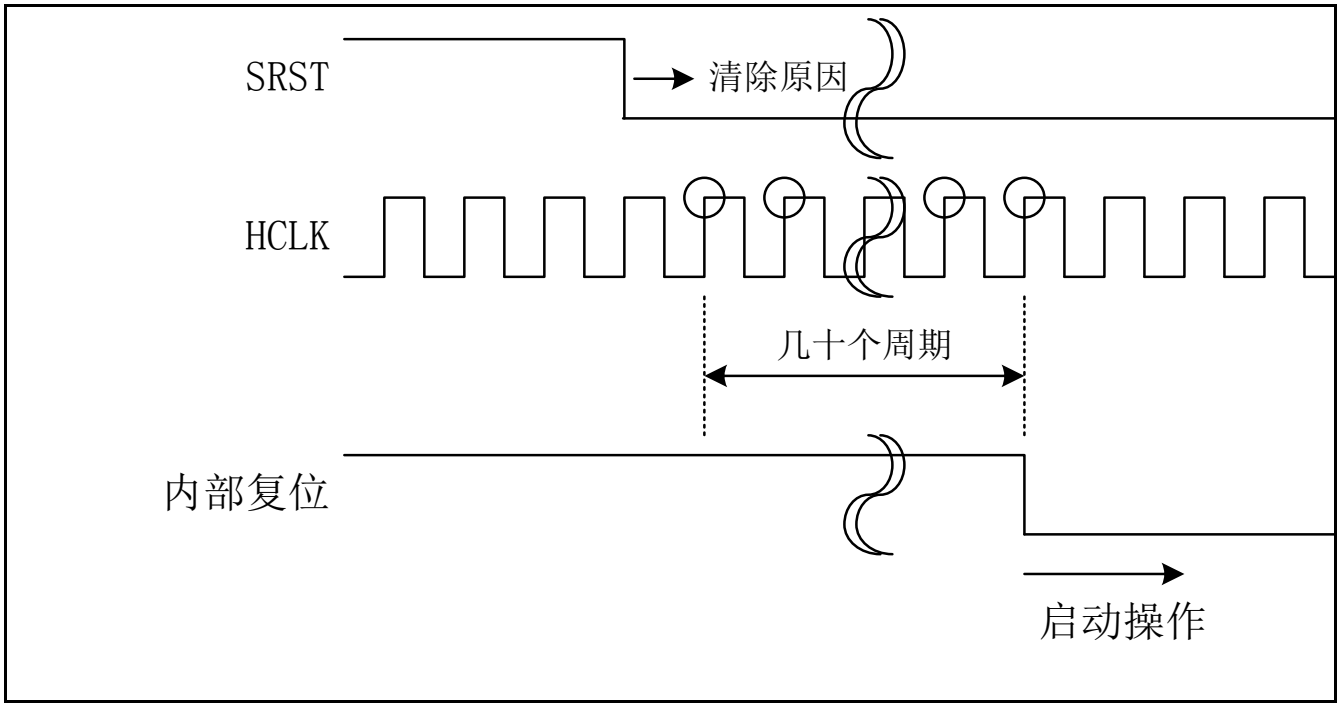
Figure 3-1 INITX 引脚输入复位被清除后的操作波形示例



SRST

Figure 3-2 所示为软件复位被清除后的操作波形示例。

Figure 3-2 软件复位被清除后的操作波形示例



4. 寄存器

本节说明复位寄存器。

复位寄存器表

Table 4-1 复位寄存器表

缩写	寄存器名称	参考章节
RST_STR	复位因素寄存器	4.1

4.1 复位因素寄存器 (RST_STR)

复位因素寄存器 (RST_STR) 展示刚出现的复位因素。RST_STR 的所有位，只能通过上电复位、低压检测复位或深度待机复位来初始化，而不能通过其它的复位来初始化。读取本寄存器将 RST_STR 的所有位清除为 “0”。

初始化后，本寄存器保存所有产过的复位因素，直到被读取。

寄存器配置

位	15	14	13	12	11	10	9	8
字段	保留							SRST
属性				-				R
初始值				-				0

位	7	6	5	4	3	2	1	0
字段	FCSR	CSVR	HWDT	SWDT	Reserved		INITX	PONR
属性	R	R	R	R	-		R	R
初始值	0	0	0	0	-		0	1

注意事项：初始值为通过上电复位、低压检测复位或深度待机复位初始化的值。

寄存器功能

[bit15:9] 保留：保留位

读取值未定义。这些位写入无效。

[bit8] SRST: 软件复位标志

指示将 "1" 写入 Cortex-M0+ 内部复位控制寄存器 (SYSRESETREQ 位) 时生成的复位。

生成软件复位时，将 SRST 设置为 (SRST = 1)。读取本寄存器将此位清除为 0。

位	描述
0	未发出软件复位。
1	已发出软件复位。

[bit7] FCSR：异常频率检测复位标志

指示检测到主振荡出现异常频率时的复位。

主振荡频率超出指定值时，将发出复位指令并设置 FCSR (FCSR = 1)。读取本寄存器将此位清除为 0。

位	说明
0	未发出异常频率检测复位。
1	已发出异常频率检测复位。

[bit6] CSVR：时钟故障检测复位标志

指示检测到主振荡或副振荡出现故障时的复位。

如果检测到停止，将发出复位指令并设置 CSVR (CSVR= 1)。读取本寄存器将此位清除为 0。

位	描述
0	未发出时钟故障检测复位。
1	已发出时钟故障检测复位。

注意事项： 有关判断主振荡或副振荡是否出现故障的方法，参见另一章节 "时钟监视器"。

[bit5] HWDT：硬件看门狗复位标志

指示从硬件看门狗计时器输入的复位。

如果计时器下溢，将发出复位指令并设置 HWDT (HWDT= 1)。读取本寄存器将此位清除为 0。

位	说明
0	未发出硬件看门狗复位。
1	已发出硬件看门狗复位。

[bit4] SWDT：软件看门狗复位标志

指示从软件看门狗计时器输入的复位。

如果计时器溢出，将发出复位指令并设置 SWDT (SWDT= 1)。读取本寄存器将此位清除为 0。

位	描述
0	未发出软件看门狗复位。
1	已发出软件看门狗复位。

[bit3:2] 保留：保留位

读取值未定义。这些位写入无效。

[bit1] INITX: INITX 引脚输入复位标志

指示外部输入的复位。

如果从外部输入复位，将设置 INITX (INITX = 1)。读取本寄存器将此位清除为 0。

位	描述
0	未发出 INITX 引脚输入复位。
1	已发出 INITX 引脚输入复位。

[bit0] PONR：上电复位标志

指示上电复位、低压检测复位或深度待机复位。

如果检测到电源上升沿、低压或深度待机转换，将发出复位指令并将 PONR 初始化为 1 (PONR = 1)。读取本寄存器将此位清除为 0。

位	描述
0	未发出上电复位、低压检测复位和深度待机复位。
1	已发出上电复位、低压检测复位或深度待机复位。

注意事项：

- *确定是否通过深度待机唤醒因素寄存器 1 和 2（WRFSR 和 WIFSR）从深度待机模式唤醒。有关详细说明，参见 "低功耗模式" 一章的 "8.5 深度待机唤醒因素寄存器 1(WRFSR)" 和 "8.6 深度待机唤醒因素寄存器 2(WIFSR)"。*

第 5-1 章：低压检测概述

本章说明低压检测电路的功能和操作。

1. 概述

代码：9AFLVD-FM0T0-C03.0

1. 概述

低压检测电路用于监视电源电压，并在电源电压低于检测电压时生成复位信号和中断信号。

低压检测电路概述

■ 低压复位电路的操作
- 本电路用于监视电源电压(VCC)，并在电源电压低于指定值时生成中断信号。
- 本电路可以选择启动和停止操作。初始状态是运行。
- 本电路可以设置检测电压（视 LVD 的类型而定）。不过，生成低压复位信号后，设定值会被初始化。因此，释放电压会变为初始值。当电源电压高于释放电压时，复位释放。
- 即使在待机模式和深度待机模式下本电路也监视电源电压。
- 在待机模式和深度待机模式下检测到电源电压下降时，本电路会生成一个复位信号。

■ 低压中断电路的操作
- 本电路用于监视电源电压(VCC)，当电源电压低于指定值时会生成一个中断信号。
- 本电路可以选择操作的启动和停止。初始状态设置为禁用。
- 本电路可以设置检测电压。
- 即使在待机模式和深度待机模式本电路下也可监视电源电压。
- 在待机模式和深度待机模式下检测到电源电压下降时，会将本电路从待机模式和深度待机模式唤醒。

注意事项：
- *当低压检测中断使能或者指定了低压检测中断的检测电压时，此电路会在低压检测电路稳定等待期结束后开始进行 VCC 电压监视。*
  有关低压检测电路的稳定等待期，参见所用产品的“数据表”。
- *如果在等待低压检测电路稳定的过程中 PCLK1 被计时器模式、RTC 模式、停止模式、深度待机 RTC 模式、深度待机停止模式或 APB1 预定标寄存器 (APBC1_PSR)停止，则本电路不执行电源电压监视。当读取到状态标志且稳定等待期结束后，修改到期望的模式。*
- *为了防止写入错误，低压检测电压控制寄存器(LVD_CTL)在初始状态下是写保护的。解除写保护模式时，应在低压检测电压保护寄存器(LVD_RLR)写入 0x1ACCE553。*

■ LVD 类型

LVD 共有 3 种类型，各种类型的设置详见 Table 1-1 所示的章节。

Table 1-1　LVD 设置参考章节

产品类型	参考章节
TYPE 1-M0+	低压检测 (TYPE 1)
TYPE 2-M0+	低压检测 (TYPE 2)
TYPE 3-M0+	低压检测 (TYPE 3)

第 5-2 章: 低压检测(TYPE 1)

本章说明低压检测电路的功能和操作。

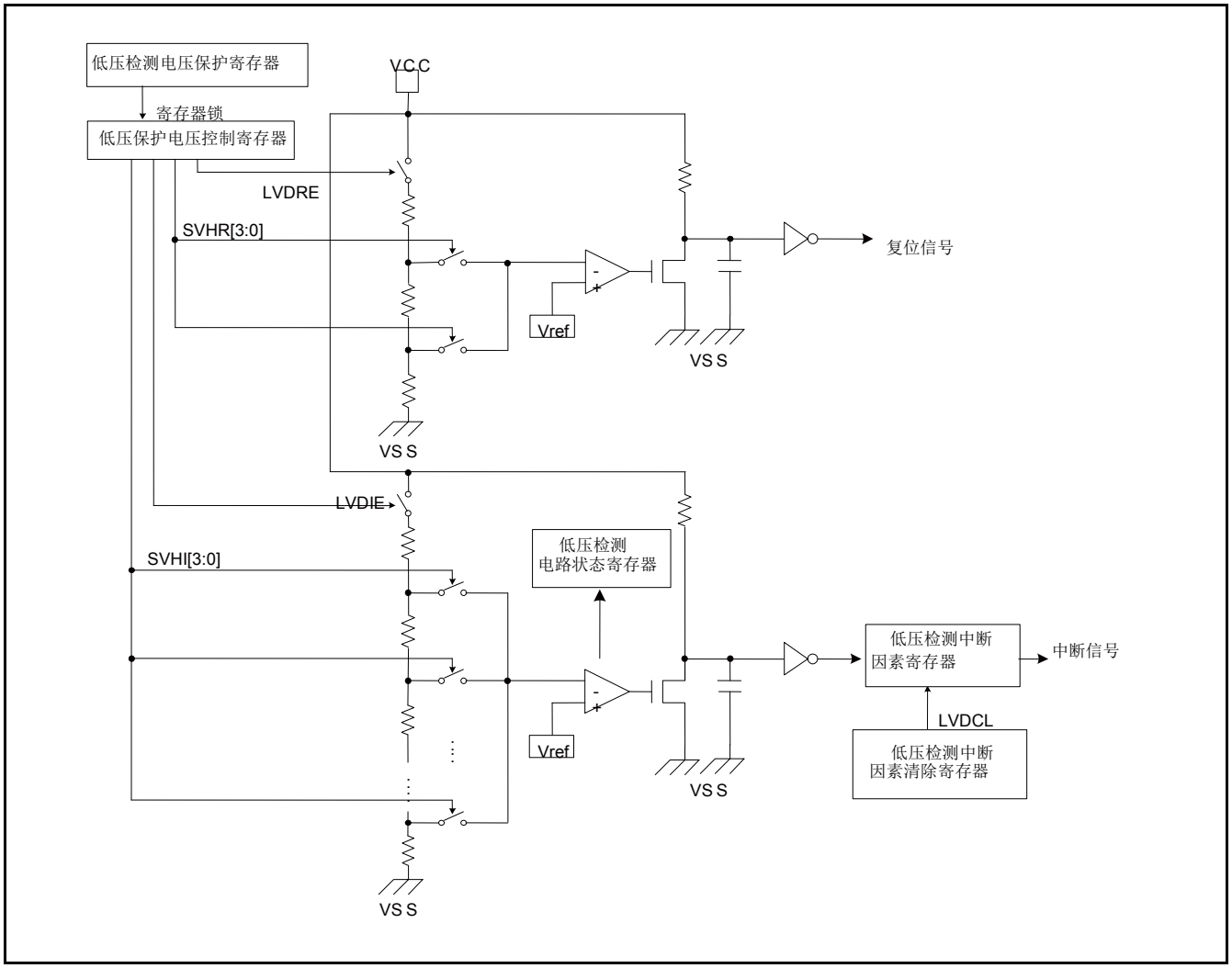
1. 配置
2. 操作
3. 设置步骤示例
4. 寄存器
5. 使用注意事项

代码： 9AFLVD-FM0T1-C01.0

1. 配置

本节描述低压检测电路的配置。

低压检测电路框图



- 低压检测电压控制寄存器 (LVD_CTL)
  此寄存器用于控制是否使能监视低压检测复位和低压检测中断的电源电压，并指定低压检测复位和低压检测中断的检测电压。
- 低压检测电压保护寄存器 (LVD_RLR)
  此寄存器用于对低压检测电压控制寄存器进行写保护。
- 低压检测中断因素寄存器 (LVD_STR)
  此寄存器用于保持低压检测中断因素。
- 低压检测中断因素清除寄存器 (LVD_CLR)
  此寄存器用于清除低压检测中断因素。

■ 低压检测电路状态寄存器(LVD_STR2)
此寄存器用于检查低压检测中断电路的操作状态。

低压检测电路引脚

以下说明低压检测电路中所用的引脚。

- VCC 引脚

  低压检测电路监视此引脚的电源电压。

- VSS 引脚

  该引脚为 GND 引脚，用作检测电源电压的基础。

2. 操作

本节说明低压检测复位电路和低压检测中断电路的操作。

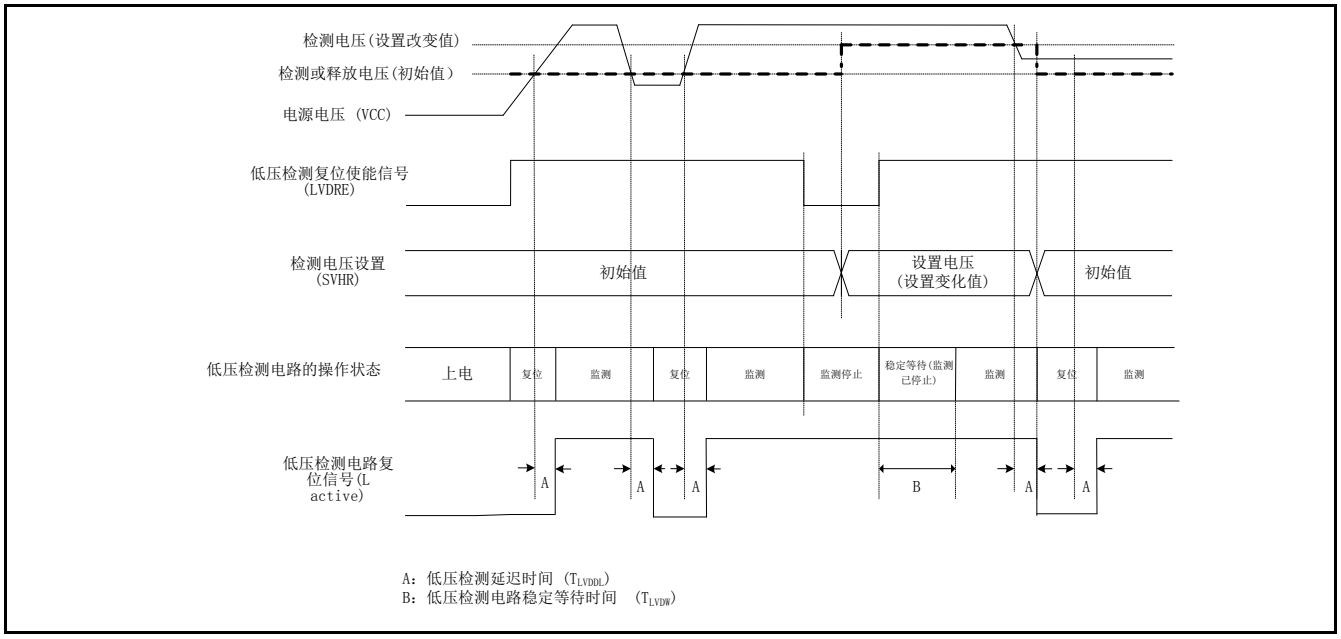
低压检测复位电路的操作

■ 电路操作说明

上电后低压检测复位电路进入监视状态。当规定的电源电压（VCC）低于检测电压时，此电路会生成一个复位信号；当电源电压超过释放电压时，复位释放。

只有当低压检测电压控制寄存器（LVD_CTL）的 LVDRE 位为"1"时复位操作才有效。复位检测电压可通过低压检测电压控制寄存器(LVD_CTL)的 SVHR 位进行设置。不过，低压检测复位会使 SVHR 位初始化。因此，释放电压会变为初始值。当电源电压高于释放电压时，复位释放。在使能了复位且设置了复位检测电压后，低压检测电路状态寄存器(LVD_STR2)的低压检测复位状态标志（LVDRRDY）变为 "1" ，并在低压检测电路的稳定等待期结束后启动电源电压监视。

本电路在待机模式（休眠模式、计时器模式、 RTC 模式及停止模式）和深度待机模式（深度待机 RTC 模式和深度待机停止模式）时有效。



低压检测中断电路的操作

■ 电路操作说明

低压检测中断电路用于监视电源电压(VCC)，并在电源电压低于指定值时生成一个中断信号。

当低压检测电压控制寄存器 (LVD_CTL)的 LVDIE 位设置为 "1"时，使能中断请求。本位的初始设置为“未使能 ("0")”。中断检测电压可通过低压检测电压控制寄存器 (LVD_CTL)的 SVHI 位进行设置。当中断请求使能且设置了中断检测电压时，低压检测电路状态寄存器(LVD_STR2)的低压检测中断状态标志(LVDIRDY)设置变为 "1"，该电路会在低压检测电路的稳定等待期结束后启动电源电压监视。

本电路在待机模式（休眠模式、计时器模式、 RTC 模式及停止模式）和深度待机模式（深度待机 RTC 模式和深度待机停止模式）下有效。当 CPU 从这些模式唤醒时也适用。

■ 低压检测中断请求

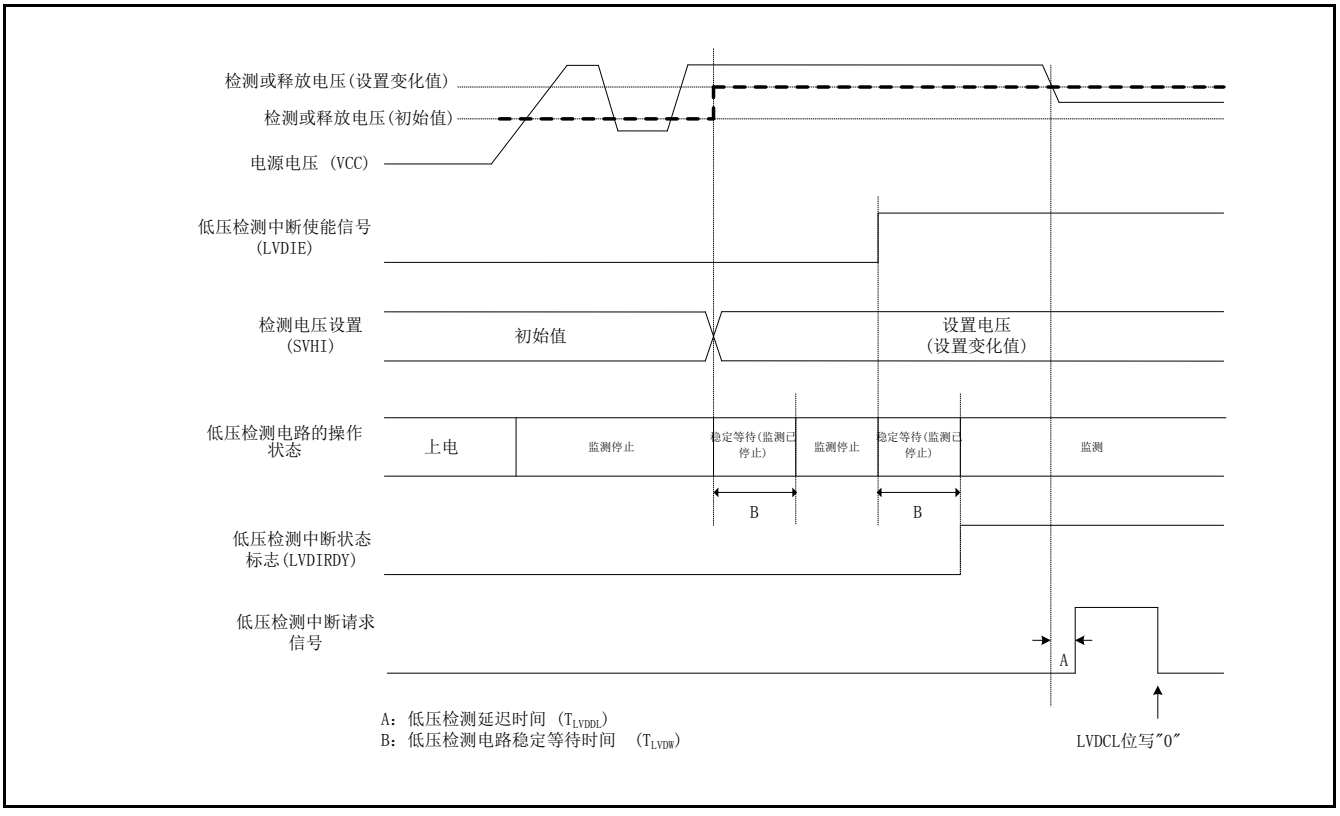
如果使能了低压检测中断功能，当电源电压 (VCC)低于指定电压时，低压检测中断因素寄存器(LVD_STR)的 LVDIR 位设置变为"1"，生成一个中断请求信号。

可通过读取 LVDIR 位来查看中断请求。

■ 取消低压检测中断请求

将低压检测中断因素寄存器(LVD_CLR)的 LVDCL 位设置为 "0" ，即可取消低压检测中断请求。这样便可清除低压检测中断因素并取消低压检测中断请求。

即使当电源电压低于指定的检测电压时，也可通过将 LVDCL 位设置为"0"来取消中断请求。



注意事项:

- *如果在等待低压检测电路稳定时 PCLK1 被计时器模式、RTC 模式、停止模式、深度待机 RTC 模式、深度待机停止模式或 APB1 预分频寄存器 (APBC1_PSR) 停止，则本电路不监视电源电压。检查低压检测电路状态寄存器(LVD_STR2)的低压检测中断状态标志(LVDIRDY)设置为"1"后，然后改变至期望模式。*

3. 设置步骤示例

本节以实例说明低压检测电路的设置步骤。

Figure3-1 低压检测复位的设置步骤示例

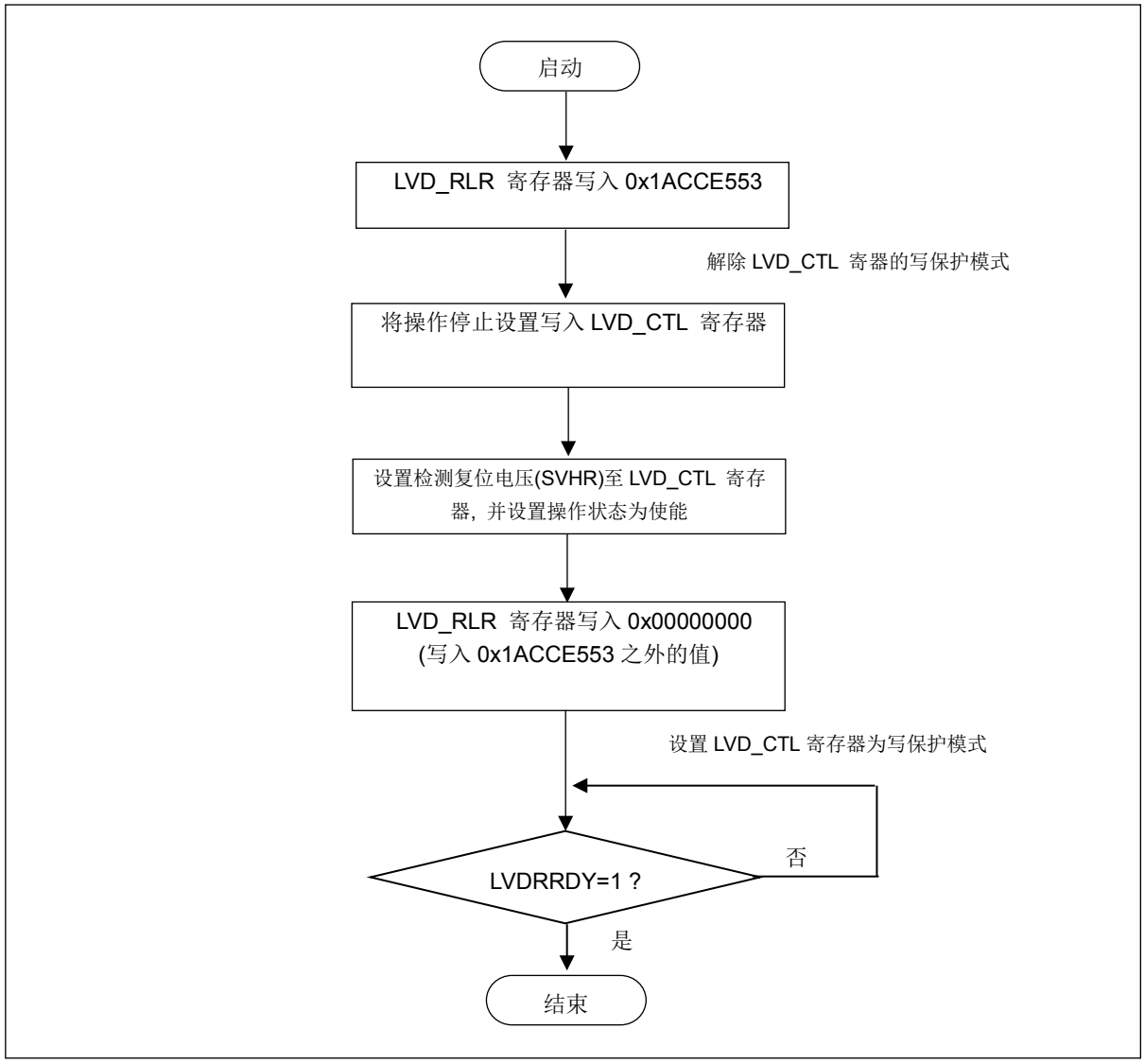
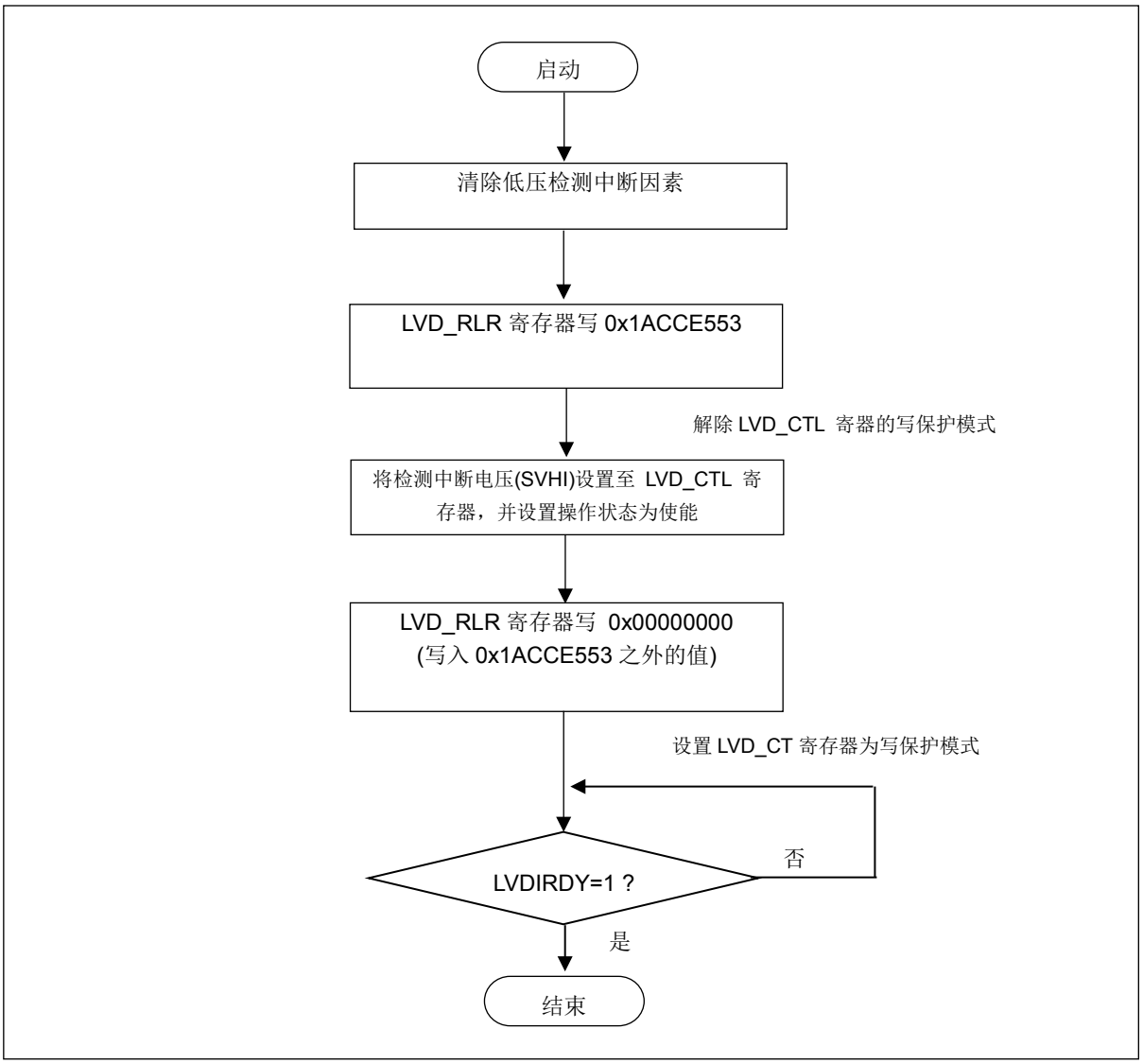


Figure3-2 低压检测中断的设置步骤示例



4. 寄存器

本节说明低压检测电路中所用寄存器的配置和功能。

低压检测电路寄存器列表

Table4-1 低压检测电路寄存器列表

缩写	寄存器名称	参考章节
LVD_CTL	低压检测电压控制寄存器	4.1
LVD_STR	低压检测中断因素寄存器	4.2
LVD_CLR	低压检测中断因素清除寄存器	4.3
LVD_RLR	低压检测电压保护寄存器	4.4
LVD_STR2	低压检测电路状态寄存器	4.5

4.1 低压检测电压控制寄存器 (LVD_CTL)

低压检测电压控制寄存器(LVD_CTL)用于控制是否使能监视低压检测复位和低压检测中断的电源电压，并指定低压检测复位和低压检测中断的检测电压。

寄存器配置

位	15	14	13	12	11	10	9	8
字段	LVDRE	SVHR					保留	
属性	R/W	R/W					-	
初始值	1	00000					00	

位	7	6	5	4	3	2	1	0
字段	LVDIE	SVHI					保留	
属性	R/W	R/W					-	
初始值	0	00011					00	

寄存器功能

[bit15] LVDRE：低压检测复位操作使能位

本位用于使能电压检测复位的电源电压监视功能。如果未使能电源电压监视，低压检测复位电路停止操作。

位	描述
0	生成低压检测复位的功能未使能。
1	生成低压检测复位的功能使能。[初始值]

[bit14:10] SVHR：低压检测复位电压设置位

这些位用于设置低压检测复位的检测电压。

bit14:10	描述
00000	将低压检测复位电压设置为 2.45 V 左右。 [初始值]
00001	将低压检测复位电压设置为 2.60 V 左右。
00010	将低压检测复位电压设置为 2.70 V 左右。
00011	将低压检测复位电压设置为 2.80 V 左右。
00100	将低压检测复位电压设置为 3.00 V 左右。
00101	将低压检测复位电压设置为 3.20 V 左右。
00110	将低压检测复位电压设置为 3.60 V 左右。
00111	将低压检测复位电压设置为 3.70 V 左右。
01000	将低压检测复位电压设置为 4.00 V 左右。
01001	将低压检测复位电压设置为 4.10 V 左右。
01010	将低压检测复位电压设置为 4.20 V 左右。
其它	禁止设置。

[bit9:8]保留：保留位
读取值总为"0"。这些位写入无效。

[bit7] LVDIE: 低压检测中断使能位
本位用于使能低压检测中断的电源电压监视。当电源电压监视未使能时，低压检测中断电路停止。

位	描述
0	低压检测中断生成功能未使能。[初始值]
1	使能低压检测中断生成功能。

[bit6:2] SVHI: 低压检测中断电压设置位
这些位用于指定低压检测中断的检测电压。

bit6:2	描述
00011	将低压检测中断电压设置为 2.80 V 左右。 [初始值]
00100	将低压检测中断电压设置为 3.00 V 左右。
00101	将低压检测中断电压设置为 3.20 V 左右。
00110	将低压检测中断电压设置为 3.60 V 左右。
00111	将低压检测中断电压设置为 3.70 V 左右。
01000	将低压检测中断电压设置为 4.00 V 左右。
01001	将低压检测中断电压设置为 4.10 V 左右。
01010	将低压检测中断电压设置为 4.20 V 左右。
其它	禁止设置。

[bit1:0]保留：保留位
读取值总为"0"。这些位写入无效。

注意事项:

- *必须在低压检测中断因素清除寄存器 (LVD_CLR)的LVDCL 位写入"0"后，使能低压检测中断使能位(LVDIE)，清除低压检测中断因素位(LVDIR)。*
- *如果低压检测中断使能位(LVDIE) 未使能，检测低压中断的低压检测电路停止工作。因此，低压检测中断因素位(LVDIR) 不设置。*
- *低压检测电压控制寄存器(LVD_CTL)在初始状态下是写保护的，因此除非解除保护模式，否则无法进行写入操作。要写入低压检测电压控制寄存器(LVD_CTL)，应将低压检测电压保护寄存器(LVD_RLR)设置为0x1ACCE553，解除保护模式。*
- *此寄存器通过上电复位和电压监视复位进行初始化。这些复位之外的其它复位因素无法对此寄存器进行初始化。*
- *低压检测复位电压的设定值被修改后，当低压检测复位使设定值初始化时，释放电压即成为初始值。这种情况下，如果电源电压高于释放电压，复位释放。检测电压和释放电压的初始值参见所用产品的数据表。*
- *深度待机转换复位不会使此寄存器初始化。*

4.2 低压检测中断因素寄存器 (LVD_STR)

低压检测中断因素寄存器 (LVD_STR) 用于保持低压检测中断因素。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	LVDIR	保留						
属性	R	-						
初始值	0	0000000						

寄存器功能

[bit7] LVDIR: 低压检测中断因素位

位	描述
0	未检测到低压检测中断请求。[初始值]
1	已检测到低压检测中断请求。

[bit6:0]保留：保留位

读取值总为"0"。这些位写入无效。

注意事项:

- *深度待机转换复位不会使此寄存器初始化。*

4.3 低压检测中断因素清除寄存器 (LVD_CLR)

低压检测中断因素清除寄存器 (LVD_CLR) 用于清除低压检测中断因素。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	LVDCL	保留						
属性	R/W	-						
初始值	1	0000000						

寄存器功能

[bit7] LVDCL: 低压检测中断因素清除位

位	描述
0	将低压检测中断因素寄存器(LVD_STR)的低压检测中断因素位(LVDIR)清除为"0"。
1	写入模式时操作无效。[初始值]

在读取模式下总设为"1"。

[bit6:0]保留：保留位

读取值总为”0”。这些位写入无效。

注意事项:

- *深度待机转换复位不会使此寄存器初始化。*

4.4 低压检测电压保护寄存器 (LVD_RLR)

低压检测电压保护寄存器(LVD_RLR) 用于对低压检测电压控制寄存器(LVD_CTL)进行写保护。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	LVDLCK[31:16]															
属性	R/W															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	LVDLCK[15:0]															
属性	R/W															
初始值	0x0001															

寄存器功能

[bit31:0] LVDLCK[31:0]: 低压检测电压控制寄存器保护位

- 设置值 0x1ACCE553 使能低压检测控制寄存器(LVD_CTL)的可写入功能（解除写保护模式）。
- 设置为 0x1ACCE553 之外的其他值，禁止写入低压检测电压控制寄存器(LVD_CTL)（使能写保护模式）。
- 当低压检测电压控制寄存器(LVD_CTL)未设置为写保护模式时，读取值为 0x00000000。
- 当低压检测电压控制寄存器(LVD_CTL)设置为写保护模式时，读为 0x00000001。

注意事项:

- *低压检测电压控制寄存器(LVD_CTL)在初始状态下是写保护的。要对 LVD_CTL 寄存器进行写入操作时，应将低压检测电压保护寄存器(LVD_RLR)设置为 0x1ACCE553，解除写保护模式。*
- *使能 LVD_CTL 寄存器的写保护模式，是将 LVD_RLR 寄存器设置为 0x1ACCE553 之外的其他值。*
- *一旦 LVD_CTL 寄存器的写保护模式被解除，LVD_CTL 寄存器将保持写保护解除状态，直到向 LVD_CTL 寄存器写入一个 0x1ACCE553 之外的其他值。*
- *深度待机转换复位不会使此寄存器初始化。*

4.5 低压检测电路状态寄存器(LVD_STR2)

低压检测电路状态寄存器(LVD_STR2)用于检查低压检测中断的操作状态。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	LVDIRDY	LVDRRDY	保留					
属性	R	R	-					
初始值	0	1	000000					

寄存器功能

[bit7] LVDIRDY: 低压检测中断状态标志

位	描述
0	稳定等待状态或监视停止状态[初始值]
1	监视状态

本位写入模式时操作无效。

[bit6] LVDRRDY: 低压检测复位状态标志

位	描述
0	稳定等待状态或监视停止状态
1	检测状态[初始值]

本位写入模式时操作无效。

[bit5:0]保留：保留位

读取值总为"0"。这些位写入无效。

注意事项:

- *深度待机转换复位不会使此寄存器初始化。*

5. 使用注意事项

本节说明低压检测电路的使用注意事项。

■ 停止模式转换的低压检测中断因素位

在低压检测中断因素位(LVD_STRLVDIR)被清除后，即使电源电压未超出检测电压；如果要再次使能中断因素，电源电压必须要高于释放电压。

但是，清除 LVDIR 之后电源电压不高于检测电压时如果转换到停止模式，低压检测电路会执行电压比较，而不失效。

所以，转换到停止模式会重新设置低压检测中断因素，从而进入中断程序。

例如，如果在低压检测的中断程序中转换到停止模式，在清除中断因素并完成中断程序后，可以很快再次设置中断因素位，重复中断程序。

为了防止检测到低压中断后在电压未超过检测电压时发生重复中断，可禁用低压检测中断使能位 (LVDIE)，避免执行该程序。

■ 低压检测复位的检测电压设置

设置低压检测复位电压后产生低压检测复位时，检测电压设定值被初始化。如在电源电压缓慢降低的情况下，电源电压超过初始设定值时，复位释放。但是，复位使程序回到初始状态，因此需要再次设置检测电压。在电源电压已经低于检测电压时，低压检测复位又被重新设置。

也就是说，会根据电源电压的变化和程序描述重复进行检测电压修改、复位、初始化（回到初始状态）、检测电压修改和复位的循环。

可采取以下措施防止在电源电压未超出检测电压时出现低压检测复位循环指令：

- 低压检测复位的设定值只采用初始值。
- 在低压检测复位之前设置低压检测中断。确定在程序开始时是否设置中断标志，并修改低压复位检测电压设定值（如有需要）。

第 5-3 章：低压检测 (TYPE 2)

本章说明低压检测电路的功能和操作。

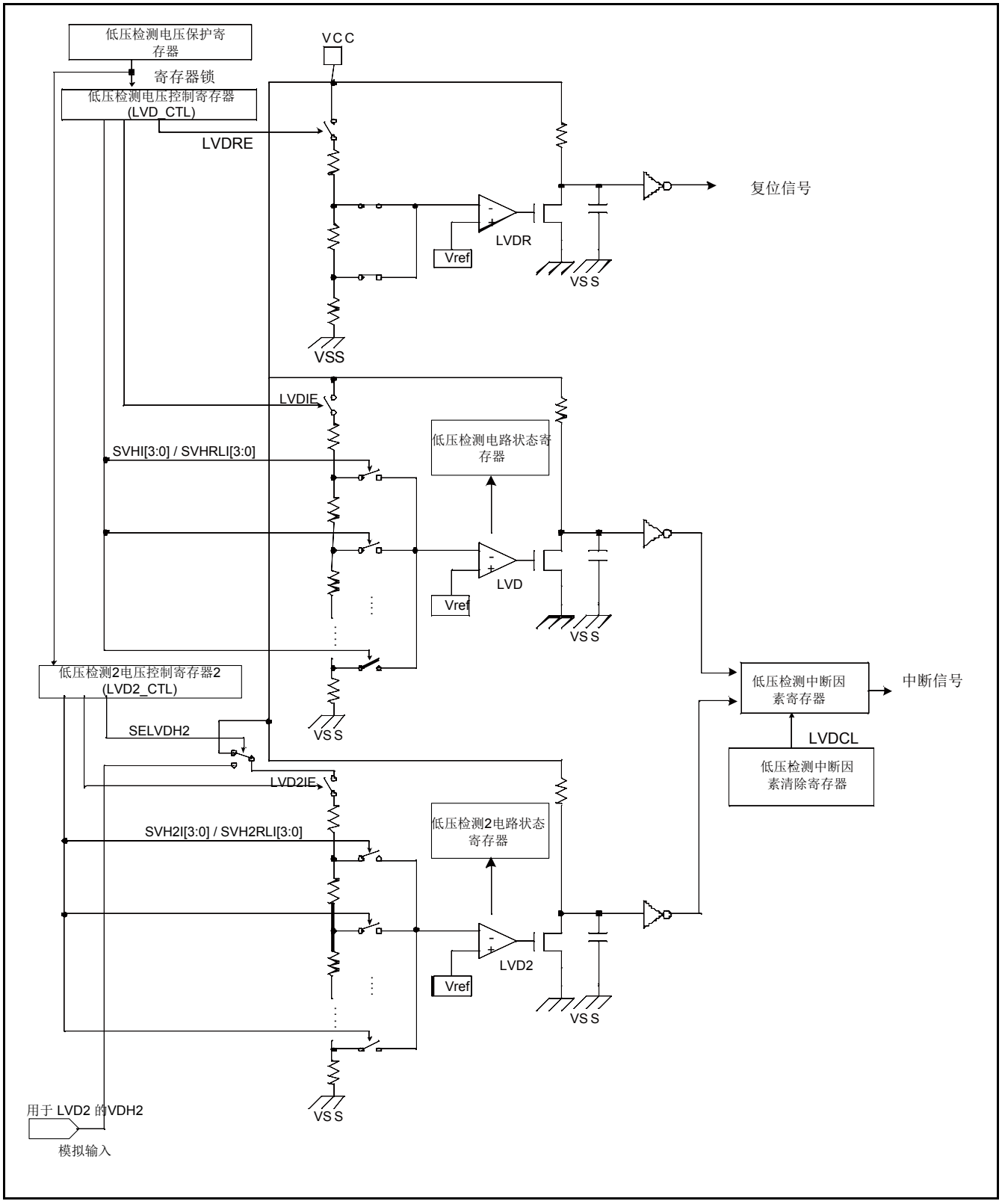
1. 配置
2. 操作
3. 设置步骤示例
4. 寄存器
5. 使用注意事项

代码：9AFLVD-FM0T1-C01.0

1. 配置

本节描述低压检测电路的配置。

低压检测电路框图


低压检测电压保护寄存器
寄存器锁
低压检测电压控制寄存器 (LVD_CTL)
LVDRE
VCC
复位信号
LVDR
Vref
VSS
LVDIE
SVHI[3:0] / SVHRLI[3:0]
低压检测电路状态寄存器
LVD
低压检测2电压控制寄存器2 (LVD2_CTL)
SELVDH2
低压检测中断因素寄存器
中断信号
LVDCL
低压检测中断因素清除寄存器
LVD2IE
SVH2I[3:0] / SVH2RLI[3:0]
低压检测2电路状态寄存器
LVD2
用于 LVD2 的VDH2
模拟输入

- 低压检测电压控制寄存器 (LVD_CTL)
  此寄存器用于控制是否使能监视低压检测复位和低压检测中断的电源电压，并指定低压检测中断的检测电压。
- 低压检测 2 电压控制寄存器 (LVD2_CTL)
  此寄存器用于控制是否使能监视低压检测中断的电源电压，并指定低压检测中断的检测电压。
- 低压检测电压保护寄存器 (LVD_RLR)
  此寄存器用于对低压检测电压控制寄存器进行写保护。
- 低压检测中断因素寄存器 (LVD_STR)
  此寄存器用于保持低压检测中断因素。
- 低压检测中断因素清除寄存器 (LVD_CLR)
  此寄存器用于清除低压检测中断因素。
- 低压检测电路状态寄存器(LVD_STR2)
  此寄存器用于检查低压检测中断电路的操作状态。
- Vref 校准控制寄存器 (CAL_CTL)
  此寄存器控制 Vref 校准。
- Vref 校准值设置寄存器(CAL_SET)
  此寄存器保持 Vref 校准值。
- Vref 校准密钥寄存器(CAL_KEY)
  此寄存器保持 CAL_CTL 和 CAL_SET 寄存器的解锁密匙。

低压检测电路引脚

以下说明低压检测电路中所用的引脚。

- VCC 引脚

  低压检测电路监视此引脚的电源电压。

- VDH2 引脚

  低压检测电路监视这个引脚的外部电压。

- VSS 引脚

  该引脚为 GND 引脚，用作检测电源电压的基础。

2. 操作

本节说明低压检测复位电路和低压检测中断电路的操作。

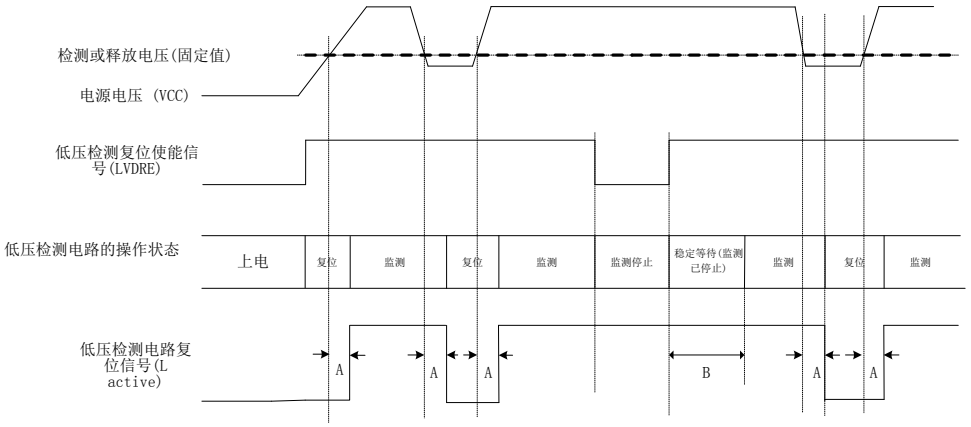
低压检测复位电路的操作

■ 电路操作说明

上电后低压检测复位电路进入监视状态。 当规定的电源电压（VCC）低于检测电压时，此电路会生成一个复位信号；当电源电压超过释放电压时，复位释放。

只有当低压检测电压控制寄存器（LVD_CTL）的 LVDRE 位为"1"时复位操作才有效。 复位检测电压为固定值，不可修改。 当电源电压高于释放电压时，复位释放。 当复位功能设置为使能后，低压检测电路状态寄存器(LVD_STR2)的低压检测复位状态标志（LVDRRDY）变为 "1" ，并在低压检测电路的稳定等待期结束后启动电源电压监测。

本电路在待机模式（休眠模式、计时器模式、 RTC 模式及停止模式）和深度待机模式（深度待机 RTC 模式和深度待机停止模式）时有效。



A：低压检测延迟时间 (T_{LVDDL})
B：低压检测电路稳定等待时间 (T_{LVDW})

低压检测中断电路的操作

■ 电路操作说明

低压检测中断电路用于监视电源电压(VCC)，并在电源电压低于指定值时生成一个中断信号。

当低压检测电压控制寄存器 (LVD_CTL)的 LVDIE 位设置为 "1"时，中断请求使能。 本位的初始设置为“未使能 ("0")”。 中断检测电压可通过低压检测电压控制寄存器(LVD_CTL)的 SVHI 位进行设置，释放电压可通过低压检测释放电压控制寄存器(LVD_CTL2)的 SVHRLI 位进行设置。 当中断请求使能且设置了中断检测/复位电压时，低压检测电路状态寄存器(LVD_STR2)的低压检测中断状态标志(LVDIRDY)设置变为 "1"，此电路会在低压检测电路的稳定等待期结束后启动电源电压监视。

本电路在待机模式（休眠模式、计时器模式、 RTC 模式及停止模式）和深度待机模式（深度待机 RTC 模式和深度待机停止模式）时有效。 同样也适用于 CPU 从这些模式唤醒时。

■ 低压检测中断请求

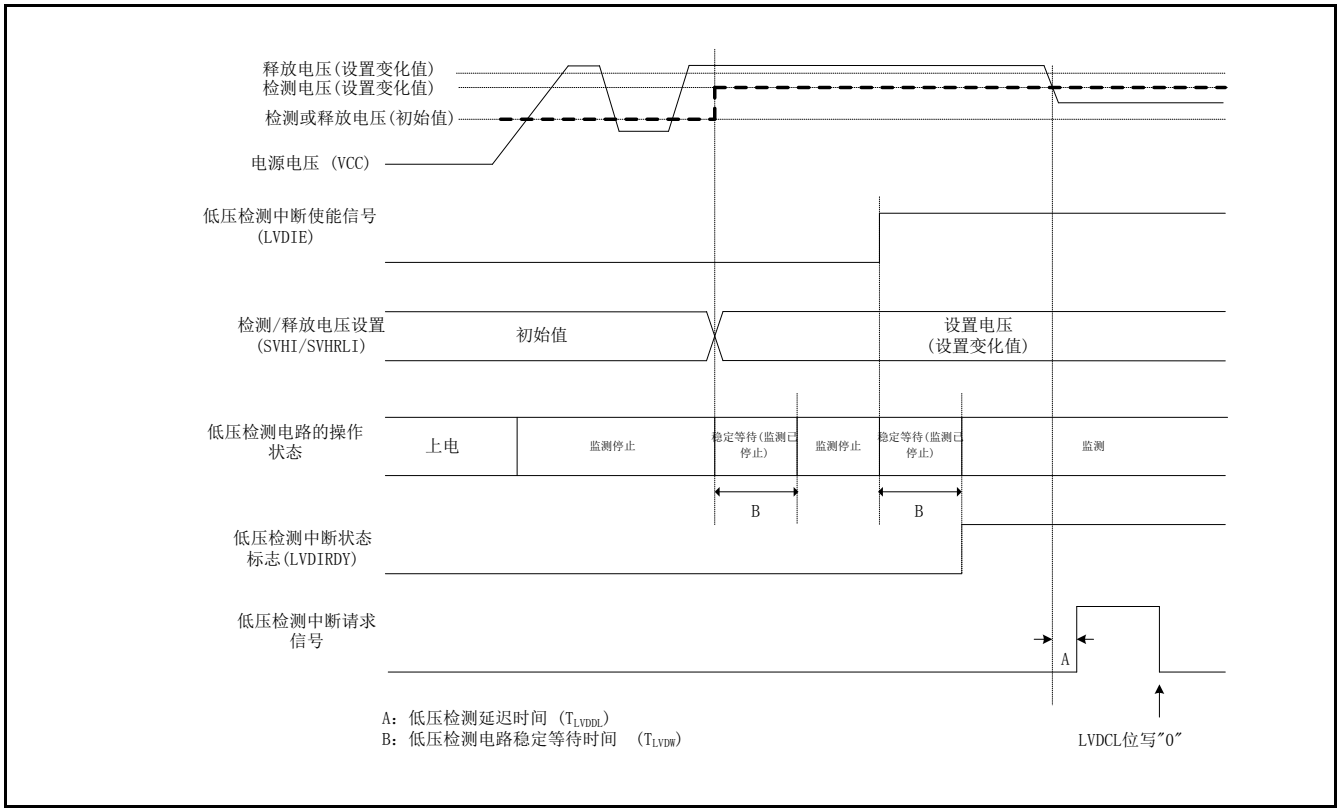
如果使能了低压检测中断功能，当电源电压 (VCC)低于指定电压时，低压检测中断因素寄存器(LVD_STR)的 LVDIR 位设置变为"1"，生成一个中断请求信号。

可通过读取 LVDIR 位来查看中断请求。

■ 取消低压检测中断请求

将低压检测中断因素寄存器(LVD_CLR)的 LVDCL 位设置为 "0" ，即可取消低压检测中断请求。 这样便可清除低压检测中断因素并取消低压检测中断请求。

即使当电源电压低于指定的检测电压时，也可通过将 LVDCL 位设置为"0"来取消中断请求。



注意事项：

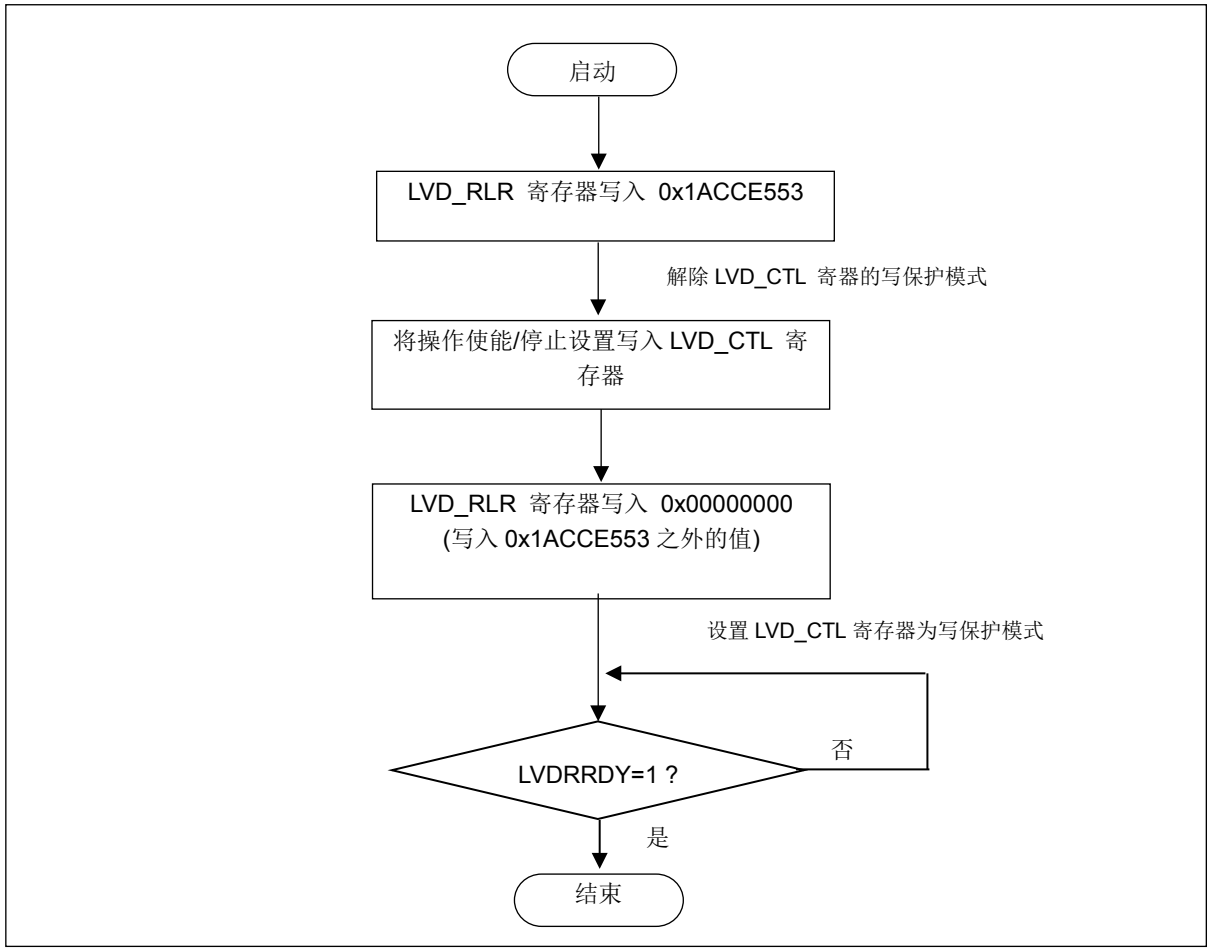
- *如果在等待低压检测电路稳定时 PCLK1 被计时器模式、RTC 模式、 停止模式、深度待机 RTC 模式、深度待机停止模式或 APB1 预定标寄存器 (APBC1_PSR) 停止，则本电路不监视电源电压 检查低压检测电路状态寄存器(LVD_STR2)的低压检测中断状态标志(LVDIRDY)设置为 "1" 后，然后改变至期望模式。*

3. 设置步骤示例

本节以实例说明低压检测电路的设置步骤。

3.1 低压检测复位示例

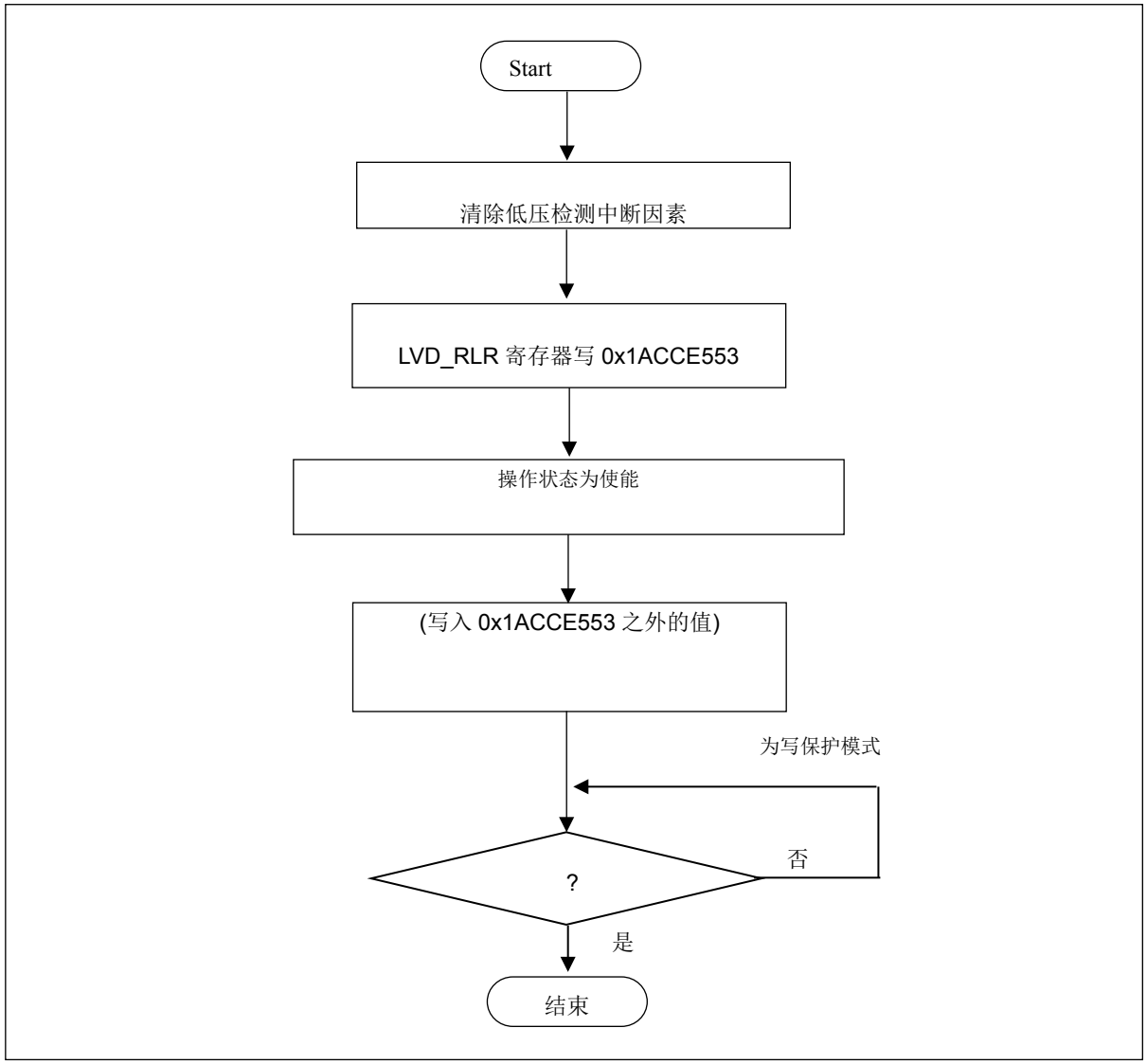
Figure 3-1 低压检测复位的设置步骤示例



3.2 低压检测中断示例 (LVDRLIE = 0)

本节说明无需单独设置释放电压的低压检测电路设置步骤。

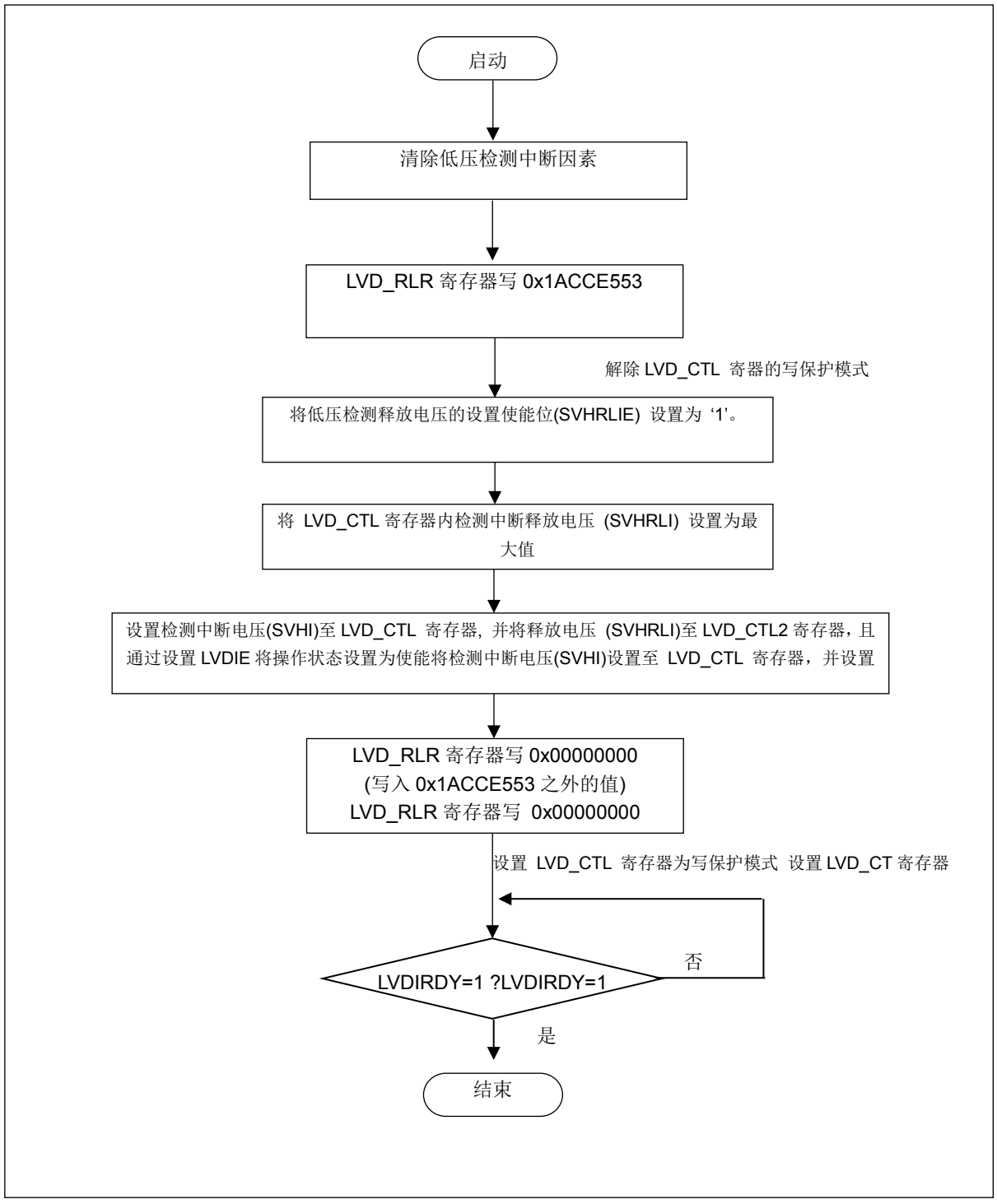
Figure 3-2 低压检测中断的设置步骤示例(LVDRLIE = 0)



3.3 低压检测中断示例 (LVDRLIE = 1)

本节说明检测电压和释放电压单独设置时的低压检测电路设置步骤。

Figure 3-3 低压检测中断的设置步骤示例(LVDRLIE = 1)

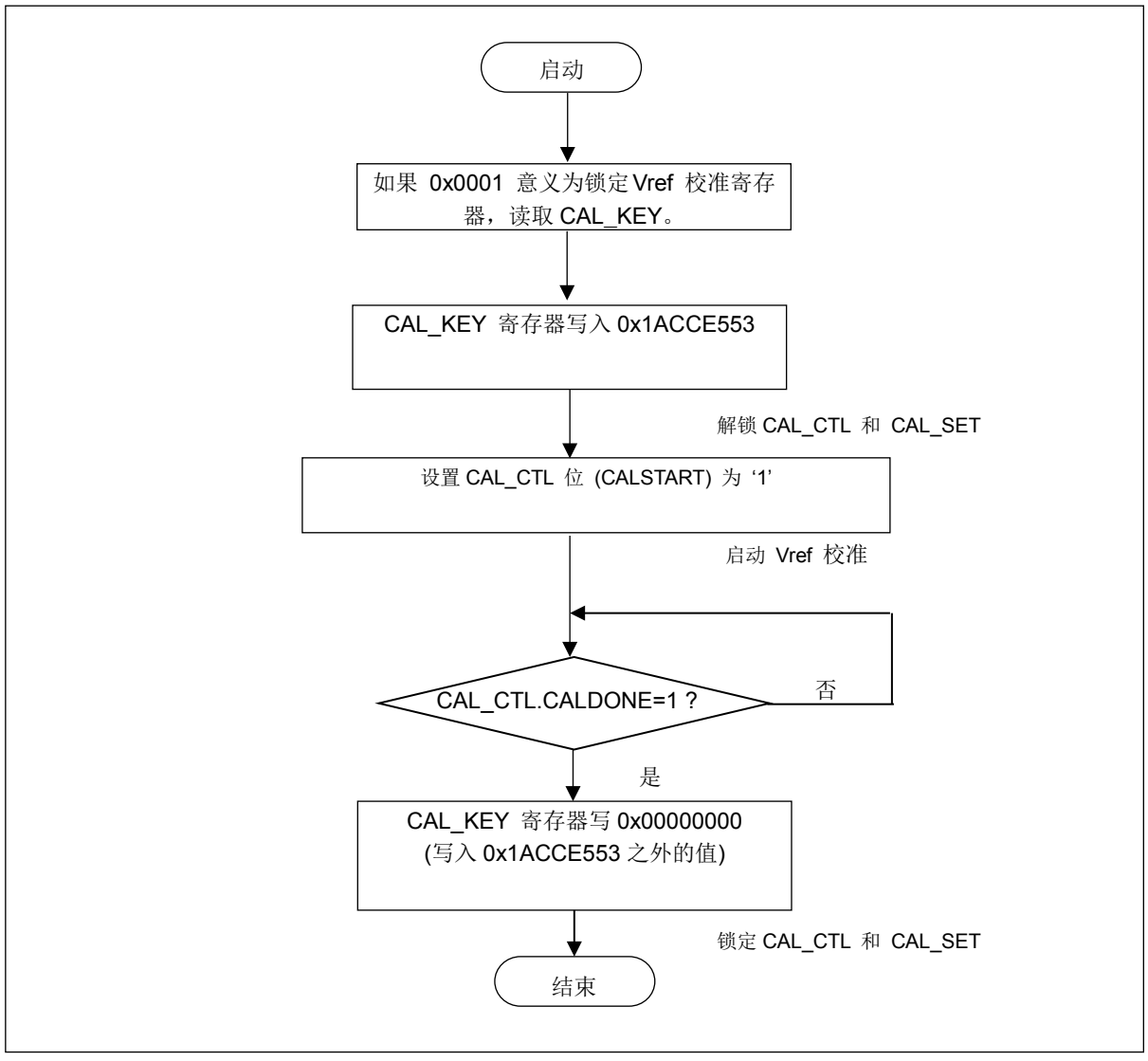


3.4 Vref 校准示例

本节说明 Vref 校准的启动步骤。

本产品配有 2 个 Vref，一个是双极 Vref，该 Vref 精确但耗电量较高；另一个是 MOS Vref，耗电量较低，经双极 Vref 校准后精确。 本部分中我们将说明如何进行 MOS Vref 校准以及如何选择 MOS Vref 校准。

Figure 3-4 Vref 校准设置步骤示例



4. 寄存器

本节说明低压检测电路中所用寄存器的配置和功能。

低压检测电路寄存器列表

Table 4-1 低压检测电路寄存器列表

缩写	寄存器名称	参考章节
LVD_CTL	低压检测电压控制寄存器	4.1
LVD_STR	低压检测中断因素寄存器	4.5
LVD_CLR	低压检测中断因素清除寄存器	4.6
LVD_RLR	低压检测电压保护寄存器	4.7
LVD_STR2	低压检测电路状态寄存器	4.8
LVD_CTL2	低压检测电压控制寄存器 2	4.2
LVD2_CTL	低压检测(ch.2)电压控制寄存器	4.3
LVD2_CTL2	低压检测(ch.2)电压控制寄存器 2	4.4
CAL_CTL	Vref 校准控制寄存器	4.9
CAL_SET	Vref 校准值寄存器	4.10
CAL_KEY	Vref 校准密钥寄存器	4.11

4.1 低压检测电压控制寄存器 (LVD_CTL)

低压检测电压控制寄存器(LVD_CTL)用于控制是否使能监视低压检测复位和低压检测中断的电源电压，并指定低压检测中断的检测电压。

寄存器配置

位	15	14	13	12	11	10	9	8
字段	LVDRE	保留					保留	
属性	R/W	-					-	
初始值	1	00000					00	

位	7	6	5	4	3	2	1	0
字段	LVDIE	SVHI					保留	
属性	R/W	R/W					-	
初始值	0	00011					00	

寄存器功能

[bit15] LVDRE：低压检测复位操作使能位

本位用于使能电压检测复位的电源电压监视功能。 如果未使能电源电压监视，低压检测复位电路停止操作。

位	描述
0	生成低压检测复位的功能未使能。
1	生成低压检测复位的功能使能。 [初始值]

[bit14:8] 保留：保留位

读取值总为"0"。这些位写入无效。

[bit7] LVDIE：低压检测中断使能位

本位用于使能低压检测中断的电源电压监视。 当电源电压监视未使能时，低压检测中断电路停止。

位	描述
0	不使能低压检测中断生成功能。 [初始值]
1	使能低压检测中断生成功能。

[bit6:2] SVHI：低压检测中断电压设置位

这些位用于指定低压检测中断的检测电压。

bit6:2	描述
00000	将低压检测中断电压设置为大约 1.50 V。
00001	将低压检测中断电压设置为大约 1.55 V。
00010	将低压检测中断电压设置为大约 1.60 V。
00011	将低压检测中断电压设置为大约 1.65 V。 [初始值]

bit6:2	描述
00100	将低压检测中断电压设置为大约 1.70 V。
00101	将低压检测中断电压设置为大约 1.75 V。
00110	将低压检测中断电压设置为大约 1.80 V。
00111	将低压检测中断电压设置为大约 1.85 V。
01000	将低压检测中断电压设置为大约 1.90 V。
01001	将低压检测中断电压设置为大约 1.95 V。
01010	将低压检测中断电压设置为大约 2.00 V。
01011	将低压检测中断电压设置为大约 2.05 V。
01100	将低压检测中断电压设置为大约 2.50 V。
01101	将低压检测中断电压设置为大约 2.60 V。
01110	将低压检测中断电压设置为大约 2.70 V。
01111	将低压检测中断电压设置为大约 2.80 V。
10000	将低压检测中断电压设置为大约 2.90 V。
10001	将低压检测中断电压设置为大约 3.00 V。
10010	将低压检测中断电压设置为大约 3.10 V。
10011	将低压检测中断电压设置为大约 3.20 V。
其它	禁止设置。

[bit1:0] 保留： 保留位

读取值总为"0"。这些位写入无效。

注意事项：

- *必须在低压检测中断因素清除寄存器 (LVD_CLR)的LVDCL 位写入"0"后，使能低压检测中断使能位(LVDIE)，清除低压检测中断因素位(LVDIR)。*
- *如果低压检测中断使能位(LVDIE) 未使能，检测低压中断的低压检测电路停止工作。 因此，低压检测中断因素位(LVDIR) 不设置。*
- *低压检测电压控制寄存器(LVD_CTL)在初始状态下是写保护的，因此除非解除保护模式，否则无法进行写入操作。 要写入低压检测电压控制寄存器(LVD_CTL)，应将低压检测电压保护寄存器(LVD_RLR)设置为0x1ACCE553，解除保护模式。*
- *此寄存器通过上电复位和电压监视复位进行初始化。 这些复位之外的其它复位因素无法对此寄存器进行初始化。*
- *低压检测复位电压为固定值。 详细信息查阅数据表。*
- *深度待机转换复位不会使此寄存器初始化。*

4.2 低压检测释放电压控制寄存器 (LVD_CTL2)

低压检测释放电压控制寄存器(LVD_CTL2)用于控制是否单独设置低压检测中断的检测电压和释放电压，并指定低压检测中断的释放电压。 此外，还可以设置 LVD 的极性。

寄存器配置

位	15	14	13	12	11	10	9	8
字段	保留							LVDPOL
属性				-				R/W
初始值				0000000				0

位	7	6	5	4	3	2	1	0
字段	LVDRLIE	SVHRLI					保留	
属性	R/W	R/W					-	
初始值	0	00011					00	

寄存器功能

[bit15:9] 保留：保留位

读取值总为"0"。这些位写入无效。

[bit8] LVDPOL：低压检测极性设置位

本位可以使 LVD 的极性反转。

详细信息参见“操作”一节。

位	描述
0	LVD 极性不反转。[初始值]
1	LVD 极性反转。

[bit7] LVDRLIE：低压检测释放电压设置使能位

本位用于确定 LVD 的检测电压和释放电压是否可以单独设置。

位	描述
0	释放电压不可设置。 SVHRLI 的值总等于 SVHI。 [初始值]
1	释放电压可单独设置。

[bit6:2] SVHRLI：低压检测释放电压设置位

这些位用于指定低压检测中断的检测电压。

bit6:2	描述
00000	将低压检测释放电压设置为大约 1.55 V。
00001	将低压检测释放电压设置为大约 1.60 V。
00010	将低压检测释放电压设置为大约 1.65 V。
00011	将低压检测释放电压设置为大约 1.70 V。 [初始值]
00100	将低压检测释放电压设置为大约 1.75 V。

bit6:2	描述
00101	将低压检测释放电压设置为大约 1.80 V。
00110	将低压检测释放电压设置为大约 1.85 V。
00111	将低压检测释放电压设置为大约 1.90 V。
01000	将低压检测释放电压设置为大约 1.95 V。
01001	将低压检测释放电压设置为大约 2.00 V。
01010	将低压检测释放电压设置为大约 2.05 V。
01011	将低压检测释放电压设置为大约 2.10 V。
01100	将低压检测释放电压设置为大约 2.60 V。
01101	将低压检测释放电压设置为大约 2.70 V。
01110	将低压检测释放电压设置为大约 2.80 V。
01111	将低压检测释放电压设置为大约 2.90 V。
10000	将低压检测释放电压设置为大约 3.00 V。
10001	将低压检测释放电压设置为大约 3.10 V。
10010	将低压检测释放电压设置为大约 3.20 V。
10011	将低压检测释放电压设置为大约 3.30 V。
其它	禁止设置。

[bit1:0] 保留： 保留位

读取值总为"0"。这些位写入无效。

注意事项：

- *使能 LVDIE 之前必须将低压检测中断释放电压设置使能位(LVDRLIE) 设置为 0。*
- *低压检测电压控制寄存器(LVD_CTL2)在初始状态下是写保护的，因此除非解除保护模式，否则无法进行写入操作。 要写入低压检测电压控制寄存器(LVD_CTL2)，应将低压检测电压保护寄存器(LVD_RLR) 设置为 0x1ACCE553，解除保护模式。*
- *此寄存器通过上电复位和电压监视复位进行初始化。 这些复位之外的其它复位因素无法对此寄存器进行初始化。*
- *当 LVDRLIE 设置为'0'时，SVHRLI 的值总等于 SVHI 的值。*
- *当 LVDRLIE=1 时，尝试将小于 SVHI 值的值写入 SVHRLI 会被拒绝。*
- *深度待机转换复位不会使此寄存器初始化。*

4.3 低压检测(ch.2)电压控制寄存器 (LVD2_CTL)

低压检测(ch.2)电压控制寄存器(LVD2_CTL)用于控制是否使能监视另一低压检测中断的电源/外部电压，并指定低压检测中断的检测电压。

寄存器配置

位	15	14	13	12	11	10	9	8
字段	保留							
属性	-							
初始值	00000000							

位	7	6	5	4	3	2	1	0
字段	LVD2IE	SVH2I					保留	
属性	R/W	R/W					-	
初始值	0	00011					00	

寄存器功能

[bit15:8] 保留：保留位

读取值总为"0"。这些位写入无效。

[bit7] LVD2IE：低压检测 2 中断使能位

本位用于使能低压检测中断（通道 2）的电源电压/外部电压监视。 当电源电压/外部电压监视未使能时，低压检测中断电路停止。

位	描述
0	不使能低压检测中断生成功能。 [初始值]
1	使能低压检测中断生成功能。

[bit6:2] SVH2I： 低压检测 2 中断电压设置位

这些位用于指定低压检测中断（通道 2）的检测电压。

bit6:2	描述
00000	将低压检测 2 中断电压设置为大约 1.50 V。
00001	将低压检测 2 中断电压设置为大约 1.55 V。
00010	将低压检测 2 中断电压设置为大约 1.60 V。
00011	将低压检测 2 中断电压设置为大约 1.65 V。 [初始值]
00100	将低压检测 2 中断电压设置为大约 1.70 V。
00101	将低压检测 2 中断电压设置为大约 1.75 V。
00110	将低压检测 2 中断电压设置为大约 1.80 V。
00111	将低压检测 2 中断电压设置为大约 1.85 V。
01000	将低压检测 2 中断电压设置为大约 1.90 V。
01001	将低压检测 2 中断电压设置为大约 1.95 V。
01010	将低压检测 2 中断电压设置为大约 2.00 V。
01011	将低压检测 2 中断电压设置为大约 2.05 V。

bit6:2	描述
01100	将低压检测 2 中断电压设置为大约 2.50 V。
01101	将低压检测 2 中断电压设置为大约 2.60 V。
01110	将低压检测 2 中断电压设置为大约 2.70 V。
01111	将低压检测 2 中断电压设置为大约 2.80 V。
10000	将低压检测 2 中断电压设置为大约 2.90 V。
10001	将低压检测 2 中断电压设置为大约 3.00 V。
10010	将低压检测 2 中断电压设置为大约 3.10 V。
10011	将低压检测 2 中断电压设置为大约 3.20 V。
其它	禁止设置。

[bit1:0] 保留： 保留位

读取值总为"0"。这些位写入无效。

注意事项：

- *必须在低压检测中断因素清除寄存器 (LVD_CLR)的LVD2CL 位写入"0"后，使能低压检测 2 中断使能位(LVD2IE)，清除低压检测中断因素位(LVD2IR)。*
- *如果低压检测 2 中断使能位(LVD2IE)未使能，检测低压中断的低压检测电路（通道 2）停止工作。因此，低压检测 2 中断因素位(LVD2IR)不设置。*
- *低压检测电压控制寄存器(LVD2_CTL)在初始状态下是写保护的，因此除非解除保护模式，否则无法进行写入操作。 要写入低压检测电压控制寄存器(LVD2_CTL)，应将低压检测电压保护寄存器(LVD_RLR)设置为 0x1ACCE553，解除保护模式。*
- *此寄存器通过上电复位和电压监视复位进行初始化。 这些复位之外的其它复位因素无法对此寄存器进行初始化。*
- *深度待机转换复位不会使此寄存器初始化。*

4.4 低压检测(ch.2)释放电压控制寄存器(LVD2_CTL2)

低压检测 2 释放电压控制寄存器(LVD2_CTL2)用于控制是否单独设置低压检测中断(ch.2)的检测电压和释放电压，并指定低压检测中断的释放电压。 此外，还可以设置 LVD2 的极性。

寄存器配置

位	15	14	13	12	11	10	9	8
字段	保留	保留						LVD2POL
属性	R/W	-						R/W
初始值	0	000000						0

位	7	6	5	4	3	2	1	0
字段	LVD2RLIE	SVH2RLI					保留	
属性	R/W	R/W					-	
初始值	0	00011					00	

寄存器功能

[bit15:9] 保留： 保留位

读取值总为"0"。这些位写入无效。

[bit8] LVD2POL： 低压检测 2 极性设置位

本位可以使 LVD2 的极性反转。

位	描述
0	LVD2 的极性不反转。[初始值]
1	LVD2 的极性反转。

[bit7] LVD2RLIE： 低压检测 2 释放电压设置使能位

本位用于确定 LVD 的检测电压和释放电压是否可以单独设置。

位	描述
0	释放电压不可设置。 SVH2RLI 的值总等于 SVH2I。 [初始值]
1	释放电压可单独设置。

[bit6:2] SVH2RLI： 低压检测释放电压设置位

这些位用于指定低压检测中断的检测电压。

bit6:2	描述
00000	将低压检测 2 释放电压设置为大约 1.55 V。
00001	将低压检测 2 释放电压设置为大约 1.60 V。
00010	将低压检测 2 释放电压设置为大约 1.65 V。
00011	将低压检测 2 释放电压设置为大约 1.70 V。 [初始值]
00100	将低压检测 2 释放电压设置为大约 1.75 V。
00101	将低压检测 2 释放电压设置为大约 1.80 V。
00110	将低压检测 2 释放电压设置为大约 1.85 V。
00111	将低压检测 2 释放电压设置为大约 1.90 V。
01000	将低压检测 2 释放电压设置为大约 1.95 V。
01001	将低压检测 2 释放电压设置为大约 2.00 V。
01010	将低压检测 2 释放电压设置为大约 2.05 V。
01011	将低压检测 2 释放电压设置为大约 2.10 V。
01100	将低压检测 2 释放电压设置为大约 2.60 V。
01101	将低压检测 2 释放电压设置为大约 2.70 V。
01110	将低压检测 2 释放电压设置为大约 2.80 V。
01111	将低压检测 2 释放电压设置为大约 2.90 V。
10000	将低压检测 2 释放电压设置为大约 3.00 V。
10001	将低压检测 2 释放电压设置为大约 3.10 V。
10010	将低压检测 2 释放电压设置为大约 3.20 V。
10011	将低压检测 2 释放电压设置为大约 3.30 V。
其它	禁止设置。

[bit1:0] 保留： 保留位

读取值总为"0"。这些位写入无效。

注意事项：

- *使能 LVDIE 之前必须将低压检测中断释放电压设置使能位(LVDRLIE)设置为 0。*
- *当 LVDRLIE 设置为'0'时，SVHRLI 的值总等于 SVHI 的值。*
- *低压检测电压控制寄存器(LVD_CTL2)在初始状态下是写保护的，因此除非解除保护模式，否则无法进行写入操作。 要写入低压检测电压控制寄存器(LVD_CTL2)，应将低压检测电压保护寄存器(LVD_RLR)设置为 0x1ACCE553，解除保护模式。*
- *此寄存器通过上电复位和电压监视复位进行初始化。 这些复位之外的其它复位因素无法对此寄存器进行初始化。*
- *尝试将小于 SVH2I 值的值写入 SVH2RLI 时，写入访问会被拒绝。*
- *深度待机转换复位不会使此寄存器初始化。*

4.5 低压检测中断因素寄存器 (LVD_STR)

低压检测中断因素寄存器 (LVD_STR) 用于保持低压检测中断因素。

寄存器配置

位	15	14	13	12	11	10	9	8
字段	LVD2IR	保留						LVD2HSTR
属性	R	-						R
初始值	0	000000						0

字段	7	6	5	4	3	2	1	0
属性	LVDIR	保留						LVDHSTR
初始值	R	-						R
字段	0	000000						0

寄存器功能

[bit15] LVD2IR： 低压检测 2 中断因素位

位	描述
0	未检测到低压检测 2 中断请求。 [初始值]
1	已检测到低压检测 2 中断请求。

[bit14:9] 保留： 保留位

读取值总为"0"。这些位写入无效。

[bit8] LVD2HSTR：

指示 LVD2 的初始输出。 不受"LVD2POL"设置的影响。

位	描述
0	VCC 的电压低于 SVH2I 的设定值。
1	VCC 的电压高于 SVH2RLI 的设定值。 [初始值]

[bit7] LVDIR： 低压检测中断因素位

位	描述
0	未检测到低压检测中断请求。 [初始值]
1	已检测到低压检测中断请求。

[bit6:1] 保留： 保留位

读取值总为"0"。这些位写入无效。

[bit0] LVDHSTR：

指示 LVDH 的初始输出。 不受"LVDPOL"设置的影响。

位	描述
0	VCC 的电压低于 SVHI 的设定值。
1	VCC 的电压高于 SVHRLI 的设定值。 [初始值]

注意事项：

- *深度待机转换复位不会使此寄存器初始化。*

4.6 低压检测中断因素清除寄存器 (LVD_CLR)

低压检测中断因素清除寄存器 (LVD_CLR) 用于清除低压检测中断因素。

寄存器配置

位	15	14	13	12	11	10	9	8
字段	LVD2CL	保留						
属性	R1/W0	-						
初始值	1	0000000						

位	7	6	5	4	3	2	1	0
字段	LVDCL	保留						
属性	R1/W0	-						
初始值	1	0000000						

寄存器功能

[bit15] LVD2CL： 低压检测 2 中断因素清除位

位	描述
0	将低压检测 2 中断因素寄存器(LVD_STR)的低压检测 2 中断因素位(LVD2IR)清除为"0"。
1	写入模式时操作无效。 [初始值]

在读取模式下总设为"1"。

[bit14:8] 保留： 保留位

读取值总为"0"。这些位写入无效。

[bit7] LVDCL： 低压检测中断因素清除位

位	描述
0	将低压检测中断因素寄存器(LVD_STR)的低压检测中断因素位(LVDIR)清除为"0"。
1	写入模式时操作无效。 [初始值]

在读取模式下总设为"1"。

[bit6:0] 保留： 保留位

读取值总为"0"。这些位写入无效。

注意事项：

- *深度待机转换复位不会使此寄存器初始化。*

4.7 低压检测电压保护寄存器 (LVD_RLR)

低压检测电压保护寄存器(LVD_RLR) 用于对低压检测电压控制寄存器(LVD_CTL)进行写保护。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	LVDLCK[31:16]															
属性	R/W															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	LVDLCK[15:0]															
属性	R/W															
初始值	0x0001															

寄存器功能

[bit31:0] LVDLCK[31:0]： 低压检测电压控制寄存器保护位

- 设置为 0x1ACCE553 时，可对低压检测电压控制寄存器(LVD_CTL)进行写入操作（解除写保护模式）。
- 设置为 0x1ACCE553 之外的其他值，禁止写入低压检测电压控制寄存器(LVD_CTL)（使能写保护模式）。
- 当低压检测电压控制寄存器(LVD_CTL)未设置为写保护模式时，读为 0x00000000。
- 当低压检测电压控制寄存器(LVD_CTL)设置为写保护模式时，读为 0x00000001。

注意事项：

- *低压检测电压控制寄存器(LVD_CTL)在初始状态下是写保护的。 要对 LVD_CTL 寄存器进行写入操作时，应将低压检测电压保护寄存器(LVD_RLR)设置为 0x1ACCE553，解除保护模式。*
- *使能 LVD_CTL 寄存器的写保护模式，是将 LVD_RLR 寄存器设置为 0x1ACCE553 之外的其他值。*
- *一旦 LVD_CTL 寄存器的写保护模式被解除，LVD_CTL 寄存器将保持写保护解除状态，直到向 LVD_CTL 寄存器写入一个 0x1ACCE553 之外的其他值为止。*
- *深度待机转换复位不会使此寄存器初始化。*

4.8 低压检测电路状态寄存器(LVD_STR2)

低压检测电路状态寄存器(LVD_STR2)用于检查低压检测中断的操作状态。

寄存器配置

位	15	14	13	12	11	10	9	8
字段	LVD2IRDY	保留						
属性	R	-						
初始值	0	0000000						

位	7	6	5	4	3	2	1	0
字段	LVDIRDY	LVDRRDY	保留					
属性	R	R	-					
初始值	0	1	000000					

寄存器功能

[bit15] LVD2IRDY： 低压检测 2 中断状态标志

位	描述
0	稳定等待状态或监视停止状态
1	检测状态[初始值]

本位写入模式时操作无效。

[bit14:8] 保留： 保留位

读取值总为"0"。这些位写入无效。

[bit7] LVDIRDY： 低压检测中断状态标志

位	描述
0	稳定等待状态或监视停止状态[初始值]
1	监视状态

本位写入模式时操作无效。

[bit6] LVDRRDY： 低压检测复位状态标志

位	描述
0	稳定等待状态或监视停止状态
1	检测状态[初始值]

本位写入模式时操作无效。

[bit5:0] 保留： 保留位
读取值总为"0"。这些位写入无效。

注意事项：

- *深度待机转换复位不会使此寄存器初始化。*

4.9 Vref 校准控制寄存器 (CAL_CTL)

Vref 校准控制寄存器(CAL_CTL)控制 Vref 校准。

本产品配有 2 个 Vref，一个是双极 Vref，该 Vref 精确但耗电量较高；另一个是 MOS Vref，耗电量较低，经双极 Vref 校准后精确。本部分中我们将说明如何进行低功率 Vref 校准以及如何选择低功率 Vref 校准。

寄存器配置

位	15	14	13	12	11	10	9	8
字段	保留							
属性					-			
初始值					00000000			

位	7	6	5	4	3	2	1	0
字段	保留				CALDONE	BGRSEL	保留	CALSTART
属性		-			R	R/W	-	R/W
初始值		0000			1	0	0	0

寄存器功能

[bit3] CALDONE： Vref 校准完成标志

位	描述
0	Vref 校准正在进行或未启动。 [初始值]
1	Vref 校准已完成。

本位写入模式时操作无效。

[bit2] BGRSEL： Vref 选择位

位	描述
0	选择双极 Vref[初始值]
1	选择 MOS Vref。 Vref 校准完成后会自动选择 MOS Vref。

[bit0] CALSTART： Vref 校准启动位

位	描述
0	Vref 校准未启动。 [初始值]
1	写入 1 启动 Vref 校准。 校准完成后，本位自动清除为 0。

[bit15:4][bit1] 保留： 保留位

读取值总为"0"。这些位写入无效。

4.10 Vref 校准值寄存器(CAL_SET)

Vref 校准值寄存器(CAL_set)用于存储 Vref 校准值。

校准值只在选中 MOS Verf 时有效。

寄存器配置

位	15	14	13	12	11	10	9	8
字段	保留							
属性	-							
初始值	00000000							

位	7	6	5	4	3	2	1	0
字段	保留			DSTB[4:0]				
属性	-			R/W				
初始值	000			10001				

[bit4:0] DSTB： Vref 校准值

Bit4:0	描述
xxxxx	Vref 校准值。 [初始值：10001，约 1.2000V] 设置 CAL_CTL.BGRSEL=1 时，DSTB[4:0]有效。

[bit15:5] 保留： 保留位

读取值总为"0"。这些位写入无效。

4.11 Vref 校准密钥寄存器(CAL_KEY)

Vref 校准密匙用于 CAL_CTL 和 CAL_SET 写入的锁定/解除锁定。

锁定时，用户不可更改 CAL_CTL 和 CAL_SET 寄存器。

只支持字访问。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	CAL_KEY[31:24]							
属性	R0/W							
初始值	00000000							

位	23	22	21	20	19	18	17	16
字段	CAL_KEY[23:16]							
属性	R0/W							
初始值	00000000							

位	15	14	13	12	11	10	9	8
字段	CAL_KEY[15:8]							
属性	R0/W							
初始值	00000000							

位	7	6	5	4	3	2	1	0
字段	CAL_KEY[7:0]							
属性	R0/W							
初始值	00000001							

[bit31:0] CAL_KEY： Vref 校准密钥值

Bit31:0	描述
xxxxxxxx	Vref 校准密钥值。 [初始值：0001] 写入 32'h1ACCE553 时解除 CAL_CTL 和 CAL_SET 的锁定。 写入其他值时锁定上述寄存器。 读得 0x0001 时表示上述寄存器处于受保护模式。 读得 0x0000 时表示上述寄存器不受保护。

5. 使用注意事项

本节说明低压检测电路的使用注意事项。

■ 停止模式转换的低压检测中断因素位

低压检测中断因素位(LVD_STRLVDIR)清除后，即使电源电压未超出检测电压，除非电源电压高于释放电压，否则中断因素不会再次被使能。

但是，清除 LVDIR 之后电源电压不高于检测电压时如果转换到停止模式，低压检测电路会执行电压比较，而不失效。

所以，转换到停止模式会重新设置低压检测中断因素，从而进入中断程序。

例如，如果在低压检测的中断程序中转换到停止模式，在清除中断因素并完成中断程序后，可以很快再次设置中断因素位，重复中断程序。

为了防止检测到低压中断后在电压未超过检测电压时发生重复中断，可禁用低压检测中断使能位 (LVDIE)，避免执行该程序。

■ 低压检测复位的检测电压设置

设置低压检测复位电压后产生低压检测复位时，检测电压设定值被初始化。 如在电源电压缓慢降低的情况下，电源电压超过初始设定值时，复位释放。 但是，复位使程序回到初始状态，因此需要再次设置检测电压。 在电源电压已经低于检测电压时，低压检测复位又被重新设置。

也就是说，会根据电源电压的变化和程序描述重复进行检测电压修改、复位、初始化（回到初始状态）、检测电压修改和复位的循环。

可采取以下措施防止在电源电压未超出检测电压时出现低压检测复位循环指令：

- 低压检测复位的设定值只采用初始值。
- 在低压检测复位之前设置低压检测中断。 确定在程序开始时是否设置中断标志，并修改低压复位检测电压设定值（如有需要）。

第 5-4 章：低压检测(TYPE 3)

本章说明低压检测电路的功能和操作。

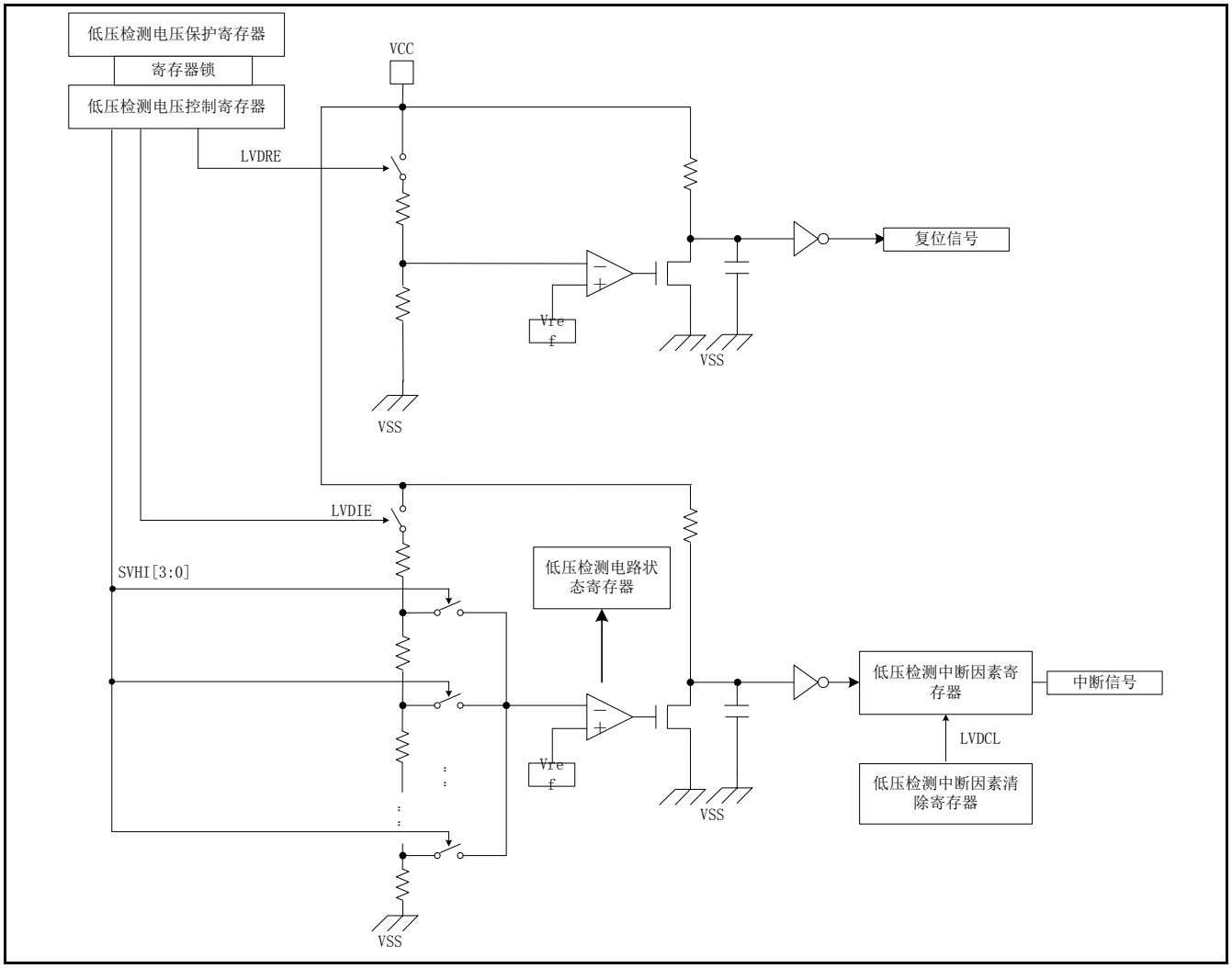
1. 配置
2. 操作
3. 设置步骤示例
4. 寄存器
5. 使用注意事项

代码： 9AFLVD-FM0T3-C03.0

1. 配置

本节描述低压检测电路的配置。

低压检测电路框图



■ 低压检测电压控制寄存器 (LVD_CTL)
此寄存器用于控制是否使能监视低压检测复位和低压检测中断的电源电压，并指定低压检测中断的检测电压。

■ 低压检测电压保护寄存器 (LVD_RLR)
此寄存器用于对低压检测电压控制寄存器进行写保护。

■ 低压检测中断因素寄存器 (LVD_STR)
此寄存器用于保持低压检测中断因素。

■ 低压检测中断因素清除寄存器 (LVD_CLR)
此寄存器用于清除低压检测中断因素。

■ 低压检测电路状态寄存器(LVD_STR2)
此寄存器用于检查低压检测中断电路的操作状态。

■ 基准电压生成电路(Vref)
Vref 是生成基准电压的电路。 本产品中配置了两种类型的 Vref，分别为双极 Vref 和 MOS Vref。 此外，双极 Vref 用于低压检测复位/中断的基准电压生成器。

低压检测电路引脚

以下说明低压检测电路中所用的引脚。

- VCC 引脚

  低压检测电路监视此引脚的电源电压。

- VSS 引脚

  该引脚为 GND 引脚，用作检测电源电压的基础。

2. 操作

本节说明低压检测复位电路和低压检测中断电路的操作。

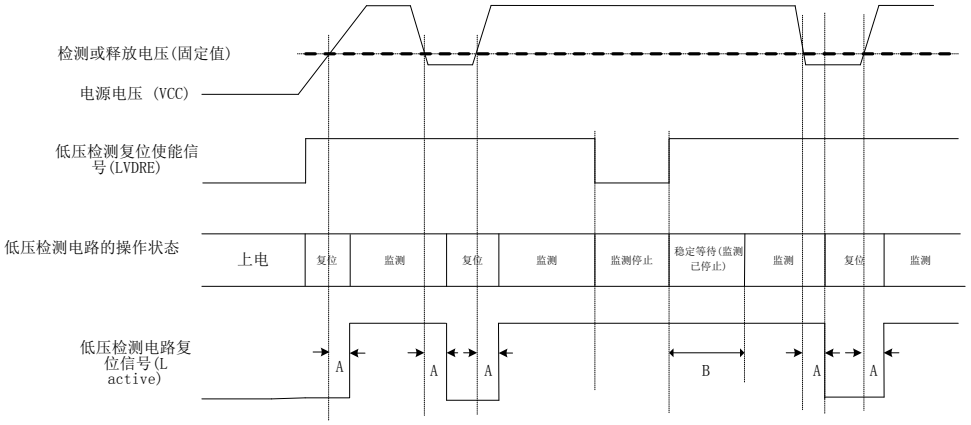
低压检测复位电路的操作

■ 电路操作说明

上电后低压检测复位电路进入监视状态。 当规定的电源电压（VCC）低于指定的检测电压时，此电路会生成一个复位信号；当电源电压超过释放电压时，复位释放。

只有当低压检测电压控制寄存器（LVD_CTL）的 LVDRE 位为"1"时复位操作才有效。 复位检测电压为固定值，不可修改。 当电源电压高于释放电压时，复位释放。 当复位功能设置为使能后，低压检测电路状态寄存器(LVD_STR2)的低压检测复位状态标志（LVDRRDY）变为 "1" ，并在低压检测电路的稳定等待期结束后启动电源电压监测。

本电路在待机模式（休眠模式、计时器模式、 RTC 模式及停止模式）和深度待机模式（深度待机 RTC 模式和深度待机停止模式）时有效。



A：低压检测延迟时间 （T_{LVDDL}）
B：低压检测电路稳定等待时间 （T_{LVDW}）

低压检测中断电路的操作

■ 电路操作说明

低压检测中断电路用于监视电源电压(VCC)，并在电源电压低于指定值时生成一个中断信号。

当低压检测电压控制寄存器 (LVD_CTL)的 LVDIE 位设置为 "1"时，中断请求使能。 本位的初始设置为“未使能 ("0")”。 中断检测电压可通过低压检测电压控制寄存器 (LVD_CTL)的 SVHI 位进行设置。 当中断请求使能且设置了中断检测电压时，低压检测电路状态寄存器(LVD_STR2)的低压检测中断状态标志(LVDIRDY)设置变为 "1"，该电路会在低压检测电路的稳定等待期结束后启动电源电压监视。

本电路在待机模式（休眠模式、计时器模式、 RTC 模式及停止模式）和深度待机模式（深度待机 RTC 模式和深度待机停止模式）时有效。 同样也适用于 CPU 从这些模式唤醒时。

■ 低压检测中断请求

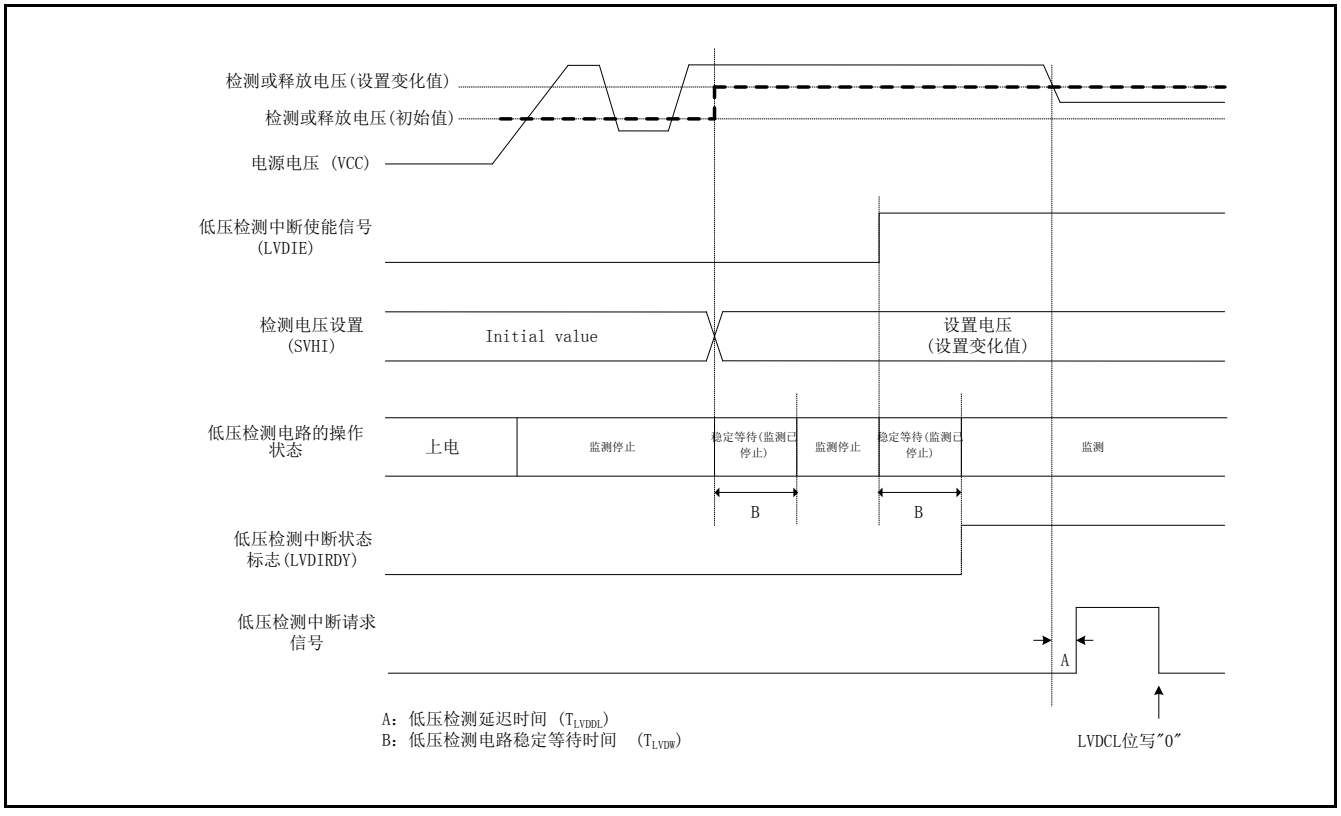
如果使能了低压检测中断功能，当电源电压 (VCC)低于指定电压时，低压检测中断因素寄存器(LVD_STR)的 LVDIR 位设置变为"1"，生成一个中断请求信号。

可通过读取 LVDIR 位来查看中断请求。

■ 取消低压检测中断请求

将低压检测中断因素寄存器(LVD_CLR)的 LVDCL 位设置为 "0" ，即可取消低压检测中断请求。 这样便可清除低压检测中断因素并取消低压检测中断请求。

即使当电源电压低于指定的检测电压时，也可通过将 LVDCL 位设置为"0"来取消中断请求。



注意事项：

- *如果在等待低压检测电路稳定时 PCLK1 被计时器模式、RTC 模式、 停止模式、深度待机 RTC 模式、深度待机停止模式或 APB1 预定标寄存器 (APBC1_PSR) 停止，则本电路不监视电源电压。检查低压检测电路状态寄存器(LVD_STR2)的低压检测中断状态标志(LVDIRDY)设置为 "1" 后，然后改变至期望模式。*

Vref 校准的操作

■ Vref 校准的目的

通常，低压检测复位/中断电路需要精确的基准电压。

本产品配置了两种类型的 Vref。 一种是双极 Vref，耗电量高但精确。 另一种是 MOS Vref，耗电量较低，但需用双极 Vref 的输出电压校准才精确。

通常，MOS Vref 是在考虑低功耗时使用。 因此，在使用 MOS Vref 之前有必要进行校准，尤其是在深度待机模式时。

■ 启动 Vref 校准

在 CAL_CTL 寄存器的 CALSTART 位写入“1”可以启动校准。 用户可通过监视 CAL_CTL 寄存器的 CALDONE 位了解校准是否完成。

校准完成后，Vref 会自动从双极 Vref 切换至 MOS Vref。

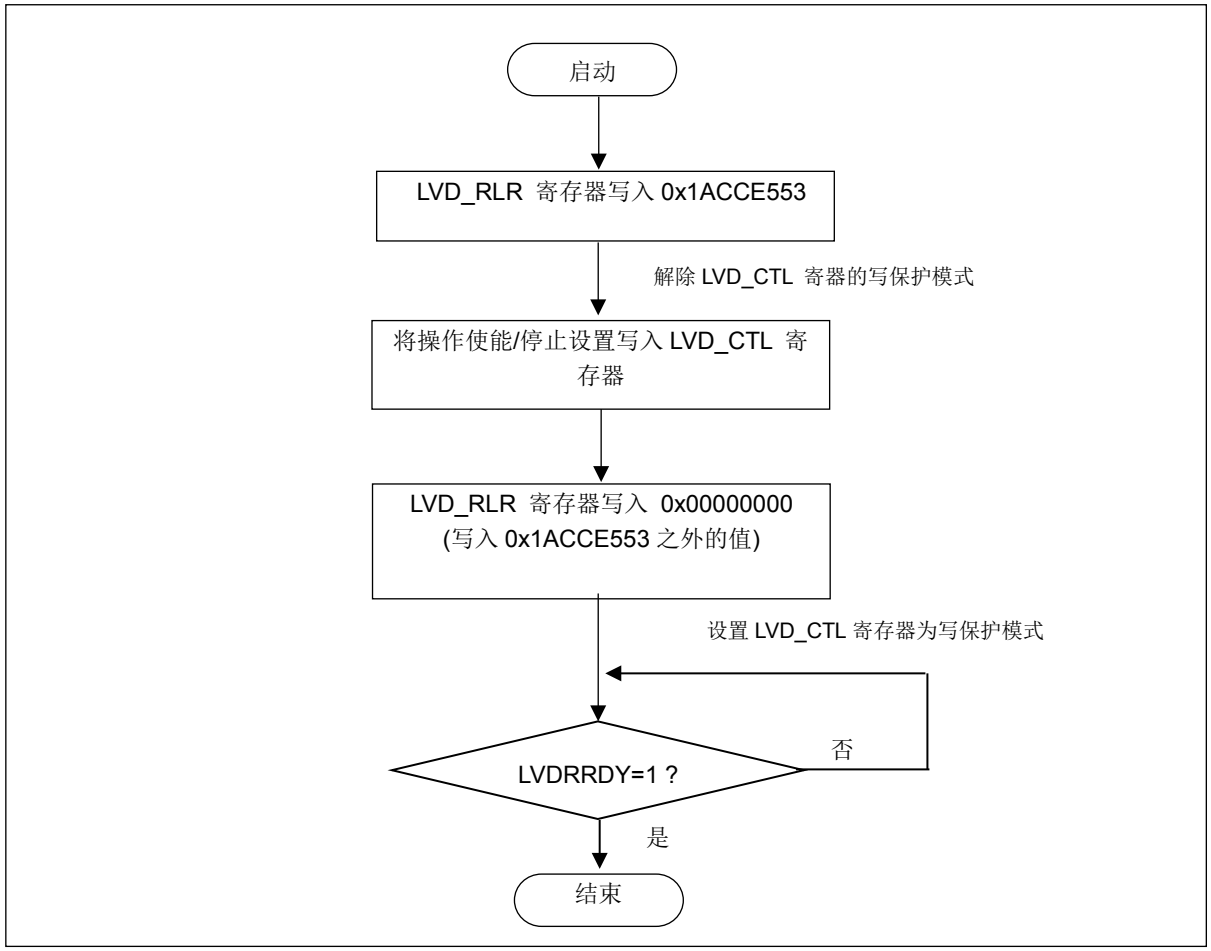
如要使用（读取/写入）CAL_CTL 寄存器，应用 CAL_KEY 寄存器进行解锁。 详细信息参见 3.3Vref 校准示例

3. 设置步骤示例

本节以实例说明低压检测电路的设置步骤。

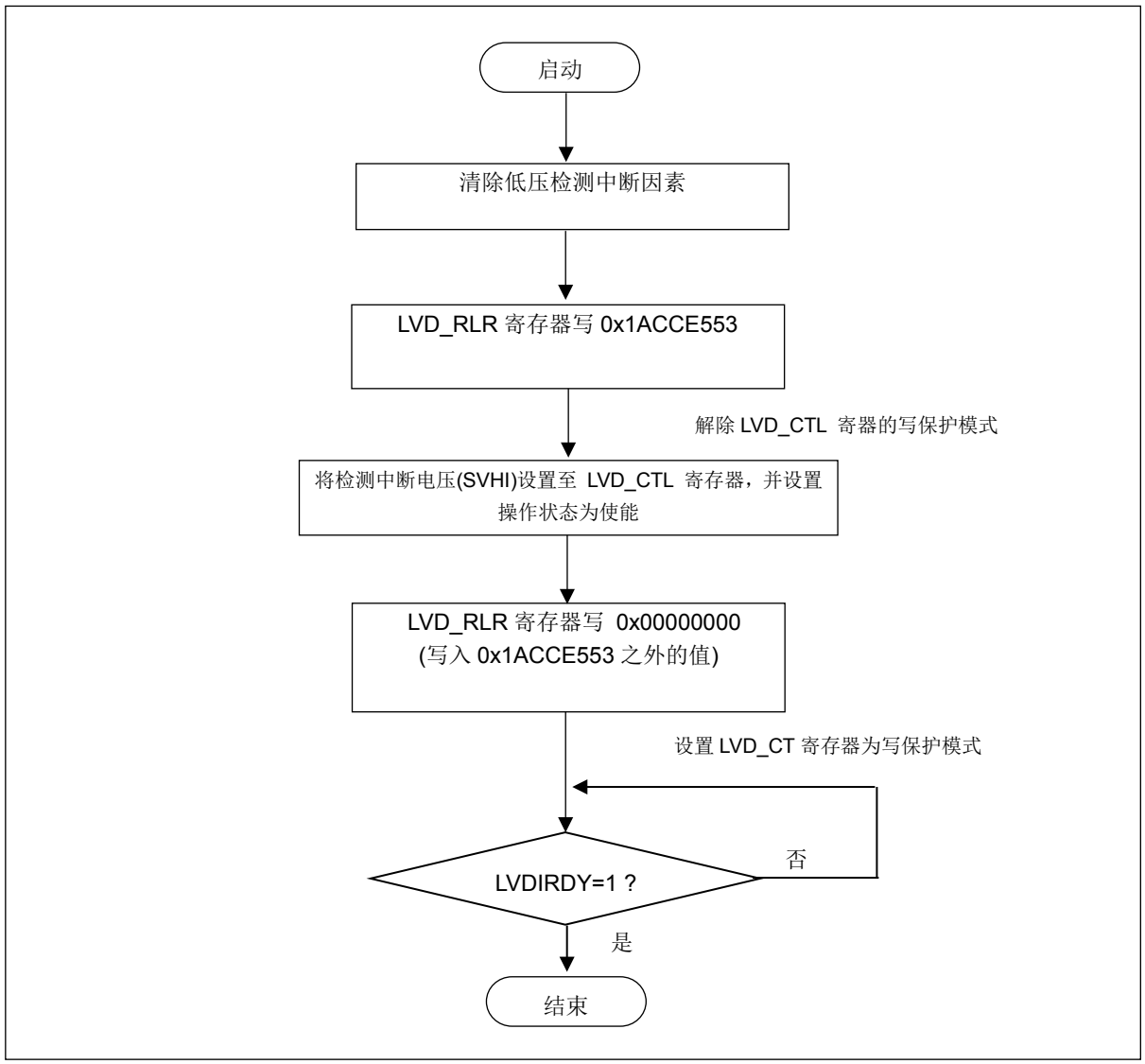
3.1 低压检测复位示例

Figure 3-1 低压检测复位的设置步骤示例



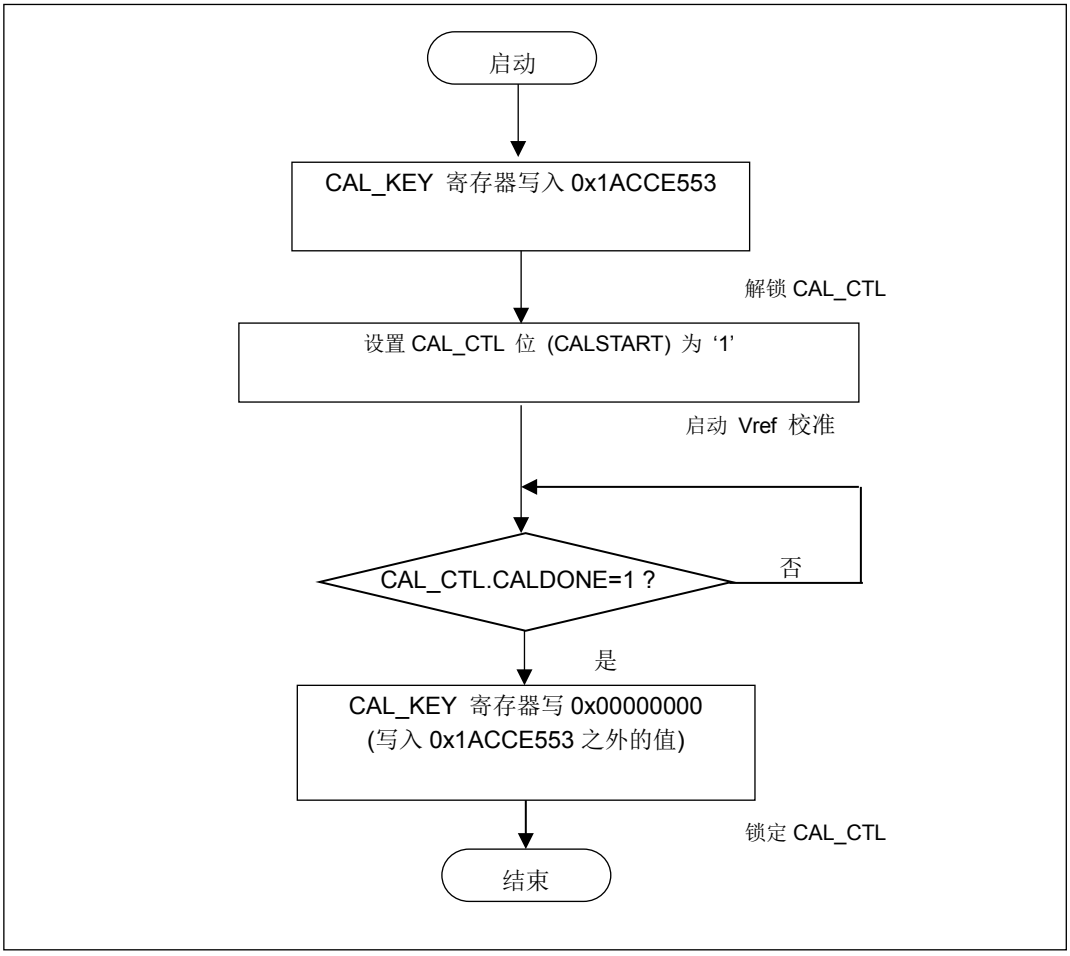
3.2 低压检测中断示例

Figure 3-2 低压检测中断的设置步骤示例



3.3 Vref 校准示例

Figure 3-3 Vref 校准的设置步骤示例



4. 寄存器

本节说明低压检测电路中所用寄存器的配置和功能。

低压检测电路寄存器列表

Table 4-1 低压检测电路寄存器列表

缩写	寄存器名称	参考章节
LVD_CTL	低压检测电压控制寄存器	4.1
LVD_STR	低压检测中断因素寄存器	4.2
LVD_CLR	低压检测中断因素清除寄存器	4.3
LVD_RLR	低压检测电压保护寄存器	4.4
LVD_STR2	低压检测电路状态寄存器	4.5
CAL_CTL	Vref 校准控制寄存器	4.6
CAL_KEY	Vref 校准密钥寄存器	4.7

4.1 低压检测电压控制寄存器 (LVD_CTL)

低压检测电压控制寄存器(LVD_CTL)用于控制是否使能监视低压检测复位和低压检测中断的电源电压，并指定低压检测中断的检测电压。

寄存器配置

位	15	14	13	12	11	10	9	8
字段	LVDRE	保留						
属性	R/W	-						
初始值	1	0000000						

位	7	6	5	4	3	2	1	0
字段	LVDIE	SVHI					保留	
属性	R/W	R/W					-	
初始值	0	00011					00	

寄存器功能

[bit15] LVDRE： 低压检测复位操作使能位

本位用于使能电压检测复位的电源电压监视功能。 如果未使能电源电压监视，低压检测复位电路停止操作。

位	描述
0	生成低压检测复位的功能未使能。
1	生成低压检测复位的功能使能。 [初始值]

[bit14:8] 保留： 保留位

读取值总为"0"。这些位写入无效。

[bit7] LVDIE： 低压检测中断使能位

本位用于使能低压检测中断的电源电压监视。 当电源电压监视未使能时，低压检测中断电路停止。

位	描述
0	不使能低压检测中断生成功能。 [初始值]
1	使能低压检测中断生成功能。

[bit6:2] SVHI： 低压检测中断电压设置位

这些位用于指定低压检测中断的检测电压。

bit6:2	描述
00000	将低压检测中断电压设置为大约 1.50 V。
00001	将低压检测中断电压设置为大约 1.55 V。
00010	将低压检测中断电压设置为大约 1.60 V。
00011	将低压检测中断电压设置为大约 1.65 V。 [初始值]

bit6:2	描述
00100	将低压检测中断电压设置为大约 1.70 V。
00101	将低压检测中断电压设置为大约 1.75 V。
00110	将低压检测中断电压设置为大约 1.80 V。
00111	将低压检测中断电压设置为大约 1.85 V。
01000	将低压检测中断电压设置为大约 1.90 V。
01001	将低压检测中断电压设置为大约 1.95 V。
01010	将低压检测中断电压设置为大约 2.00 V。
01011	将低压检测中断电压设置为大约 2.05 V。
01100	将低压检测中断电压设置为大约 2.50 V。
01101	将低压检测中断电压设置为大约 2.60 V。
01110	将低压检测中断电压设置为大约 2.70 V。
01111	将低压检测中断电压设置为大约 2.80 V。
10000	将低压检测中断电压设置为大约 2.90 V。
10001	将低压检测中断电压设置为大约 3.00 V。
10010	将低压检测中断电压设置为大约 3.10 V。
10011	将低压检测中断电压设置为大约 3.20 V。
其它	禁止设置。

[bit1:0] 保留： 保留位

读取值总为"0"。这些位写入无效。

注意事项：

- *必须在低压检测中断因素清除寄存器 (LVD_CLR)的LVDCL 位写入"0"后，使能低压检测中断使能位(LVDIE)，清除低压检测中断因素位(LVDIR)。*
- *如果低压检测中断使能位(LVDIE) 未使能，检测低压中断的低压检测电路停止工作。 因此，低压检测中断因素位(LVDIR) 不设置。*
- *低压检测电压控制寄存器(LVD_CTL)在初始状态下是写保护的，因此除非解除保护模式，否则无法进行写入操作。 要写入低压检测电压控制寄存器(LVD_CTL)，应将低压检测电压保护寄存器(LVD_RLR)设置为0x1ACCE553，解除保护模式。*
- *此寄存器通过上电复位和电压监视复位进行初始化。 这些复位之外的其它复位因素无法对此寄存器进行初始化。*
- *低压检测复位电压为固定值。 详细信息查阅数据表。*
- *深度待机转换复位不会使此寄存器初始化。*

4.2 低压检测中断因素寄存器 (LVD_STR)

低压检测中断因素寄存器 (LVD_STR) 用于保持低压检测中断因素。

寄存器配置

位	15	14	13	12	11	10	9	8
字段	保留							
属性	-							
初始值	00000000							

位	7	6	5	4	3	2	1	0
字段	LVDIR	保留						保留
属性	R	-						-
初始值	0	000000						-

寄存器功能

[bit15:8] 保留： 保留位

读取值总为"0"。这些位写入无效。

[bit7] LVDIR： 低压检测中断因素位

位	描述
0	未检测到低压检测中断请求。 [初始值]
1	已检测到低压检测中断请求。

[bit6:1] 保留： 保留位

读取值总为"0"。这些位写入无效。

[bit0] 保留： 保留位

读取值为"0" 或 “1”。这些位写入无效。

注意事项：

- *深度待机转换复位不会使此寄存器初始化。*

4.3 低压检测中断因素清除寄存器 (LVD_CLR)

低压检测中断因素清除寄存器 (LVD_CLR) 用于清除低压检测中断因素。

寄存器配置

位	15	14	13	12	11	10	9	8
字段	保留							
属性	-							
初始值	00000000							

位	7	6	5	4	3	2	1	0
字段	LVDCL	保留						
属性	R/W	-						
初始值	1	0000000						

寄存器功能

[bit15:8] 保留： 保留位

读取值总为"0"。这些位写入无效。

[bit7] LVDCL： 低压检测中断因素清除位

位		描述
写入	0	将低压检测中断因素寄存器(LVD_STR)的低压检测中断因素位(LVDIR)清除为"0"。
	1	写入模式时操作无效。 [初始值]
读取	1	总是读为 1

在读取模式下总设为"1"。

[bit6:0] 保留： 保留位

读取值总为"0"。这些位写入无效。

注意事项：

- *深度待机转换复位不会使此寄存器初始化。*

4.4 低压检测电压保护寄存器 (LVD_RLR)

低压检测电压保护寄存器(LVD_RLR) 用于对低压检测电压控制寄存器(LVD_CTL)进行写保护。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	LVDLCK[31:16]															
属性	R/W															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	LVDLCK[15:0]															
属性	R/W															
初始值	0x0001															

寄存器功能

[bit31:0] LVDLCK[31:0]： 低压检测电压控制寄存器保护位

- 设置为 0x1ACCE553 时，可对低压检测电压控制寄存器(LVD_CTL)进行写入操作（解除写保护模式）。
- 设置为 0x1ACCE553 之外的其他值，禁止写入低压检测电压控制寄存器(LVD_CTL)（使能写保护模式）。
- 当低压检测电压控制寄存器(LVD_CTL)未设置为写保护模式时，读为 0x00000000。
- 当低压检测电压控制寄存器(LVD_CTL)设置为写保护模式时，读为 0x00000001。

注意事项：

- *低压检测电压控制寄存器(LVD_CTL)在初始状态下是写保护的。 要对 LVD_CTL 寄存器进行写入操作时，应将低压检测电压保护寄存器(LVD_RLR)设置为 0x1ACCE553，解除保护模式。*
- *使能 LVD_CTL 寄存器的写保护模式，是将 LVD_RLR 寄存器设置为 0x1ACCE553 之外的其他值。*
- *一旦 LVD_CTL 寄存器的写保护模式被解除，LVD_CTL 寄存器将保持写保护解除状态，直到向 LVD_CTL 寄存器写入一个 0x1ACCE553 之外的其他值为止。*
- *深度待机转换复位不会使此寄存器初始化。*

4.5 低压检测电路状态寄存器(LVD_STR2)

低压检测电路状态寄存器(LVD_STR2)用于检查低压检测中断的操作状态。

寄存器配置

位	15	14	13	12	11	10	9	8
字段	保留							
属性	-							
初始值	00000000							

位	7	6	5	4	3	2	1	0
字段	LVDIRDY	LVDRRDY	保留					
属性	R	R	-					
初始值	0	1	000000					

寄存器功能

[bit15:8] 保留： 保留位

读取值总为"0"。这些位写入无效。

[bit7] LVDIRDY： 低压检测中断状态标志

位	描述
0	稳定等待状态或监视停止状态[初始值]
1	监视状态

本位写入模式时操作无效。

[bit6] LVDRRDY： 低压检测复位状态标志

位	描述
0	稳定等待状态或监视停止状态
1	检测状态[初始值]

本位写入模式时操作无效。

[bit5:0] 保留： 保留位

读取值总为"0"。这些位写入无效。

注意事项：

- *深度待机转换复位不会使此寄存器初始化。*

4.6 Vref 校准控制寄存器 (CAL_CTL)

Vref 校准控制寄存器(CAL_CTL)控制 Vref 校准。

寄存器配置

位	15	14	13	12	11	10	9	8
字段	保留							
属性	-							
初始值	00000000							

位	7	6	5	4	3	2	1	0
字段	保留				CALDONE	保留	保留	CALSTART
属性	-				R	-	-	R/W
初始值	0000				0	0	0	0

寄存器功能

[bit15:4] 保留： 保留位

读取值总为"0"。这些位写入无效。

[bit3] CALDONE： Vref 校准完成标志

本位指示校准是已经完成还是正在进行。

本位写入模式时操作无效。

位	描述
0	Vref 校准正在进行或未启动。 [初始值]
1	Vref 校准已完成。

[bit2] 保留： 保留位

读取值为"0" 或 “1”。写入本位时，先读取本位的值然后再重新写入相同的值。

[bit1] 保留： 保留位

读取值为"0" 或 “1”。写入本位时，先读取本位的值然后再重新写入相同的值。

[bit0] CALSTART： Vref 校准启动位

本位启动 Vref 校准。

位	描述
0	Vref 校准未启动。 [初始值]
1	写入 1 启动 Vref 校准。 校准完成后，本位自动清除为 0。

注意事项:

- *校准完成后，Vref 会从自动双极 Vref 切换至 MOS Vref。*
- *只有在上电复位后方可进行校准。 芯片正在运行时，写入此寄存器不得超过一次。*

4.7 Vref 校准密钥寄存器(CAL_KEY)

Vref 校准密匙用于 CAL_CTL 写入的锁定/解除锁定。

锁定时，用户不可更改 CAL_CTL 寄存器。

只支持字访问。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	CAL_KEY[31:24]							
属性	R0/W							
初始值	00000000							

位	23	22	21	20	19	18	17	16
字段	CAL_KEY[23:16]							
属性	R0/W							
初始值	00000000							

位	15	14	13	12	11	10	9	8
字段	CAL_KEY[15:8]							
属性	R0/W							
初始值	00000000							

位	7	6	5	4	3	2	1	0
字段	CAL_KEY[7:0]							
属性	R0/W							
初始值	00000001							

寄存器功能

[bit31:0] CAL_KEY： Vref 校准密钥位

- 设置为 0x1ACCE553 时，可对 Vref 校准控制寄存器(CAL_CTL)进行写入操作（解除写保护模式）。
- 设置为 0x1ACCE553 之外的其他值时，禁止写入 Vref 校准控制寄存器(CAL_CTL)（使能写保护模式）。
- 当 Vref 校准控制寄存器 (CAL_CTL)未设置为写保护模式时，读为 0x00000000。
- 当 Vref 校准控制寄存器 (CAL_CTL)设置为写保护模式时，读为 0x00000001。

注意事项：

- *Vref 校准控制寄存器(CAL_CTL)在初始状态下是写保护的。要写入 CAL_CTL 寄存器时，应将 Vref 密匙寄存器(CAL_KEY)设置为 0x1ACCE553 解除写保护模式。*
- *使能 CAL_CTL 寄存器的写保护模式，是将 CAL_KEY 寄存器设置为 0x1ACCE553 之外的其他值。*
- *一旦 CAL_CTL 寄存器的写保护模式被解除，CAL_CTL 寄存器将保持写保护解除状态，直到向 CAL_CTL 寄存器写入一个 0x1ACCE553 之外的其他值为止。*
- *深度待机转换复位不会使此寄存器初始化。*

5. 使用注意事项

本节说明低压检测电路的使用注意事项。

■ 停止模式转换的低压检测中断因素位

低压检测中断因素位(LVD_STRLVDIR)清除后，即使电源电压未超出检测电压，除非电源电压高于释放电压，否则中断因素不会再次被使能。

但是，清除 LVDIR 之后电源电压不高于检测电压时如果转换到停止模式，低压检测电路会执行电压比较，而不失效。

所以，转换到停止模式会重新设置低压检测中断因素，从而进入中断程序。

例如，如果在低压检测的中断程序中转换到停止模式，在清除中断因素并完成中断程序后，可以很快再次设置中断因素位，重复中断程序。

为了防止检测到低压中断后在电压未超过检测电压时发生重复中断，可禁用低压检测中断使能位 (LVDIE)，避免执行该程序。

■ 低压检测复位的检测电压设置

设置低压检测复位电压后产生低压检测复位时，检测电压设定值被初始化。 如在电源电压缓慢降低的情况下，电源电压超过初始设定值时，复位释放。 但是，复位使程序回到初始状态，因此需要再次设置检测电压。 在电源电压已经低于检测电压时，低压检测复位又被重新设置。

也就是说，会根据电源电压的变化和程序描述重复进行检测电压修改、复位、初始化（回到初始状态）、检测电压修改和复位的循环。

可采取以下措施防止在电源电压未超出检测电压时出现低压检测复位循环指令：

- 低压检测复位的设定值只采用初始值。
- 在低压检测复位之前设置低压检测中断。 确定在程序开始时是否设置中断标志，并修改低压复位检测电压设定值（如有需要）。

第 6-1 章：低功耗模式

本章说明低功耗模式的功能及操作

1. 概述
2. CPU 操作模式配置
3. 待机模式的操作
4. 待机模式设置步骤示例
5. 深度待机模式操作说明
6. 深度待机模式设置步骤示例
7. 深度待机唤醒因素确定步骤
8. 寄存器
9. 使用注意事项

代码：9AFLPMODE-FM0-C03.0

1. 概述

为了降低功耗，本系统提供低功耗模式，包括休眠待机模式、计时器模式、RTC 模式和停止模式，以及深度待机 RTC 和深度待机停止模式。

CPU 操作模式概述

CPU 操作模式分为以下类型：

- 运行模式
  - 高速 CR 运行模式
  - 主运行模式
  - PLL 运行模式
  - 低速 CR 运行模式
  - 子运行模式
- 待机模式
  - 休眠模式
    - 高速 CR 休眠模式
    - 主休眠模式
    - PLL 休眠模式
    - 低速 CR 休眠模式
    - 副休眠模式
  - 计时器模式
    - 高速 CR 计时器模式
    - 主计时器模式
    - PLL 计时器模式
    - 低速 CR 计时器模式
    - 副计时器模式
  - RTC 模式
  - 停止模式
- 深度待机模式
  - 深度待机 RTC 模式
  - 深度待机停止模式

各类型低功耗模式列表

Table 1-1 低功耗模式列表

模式	TYPE 1-M0+	TYPE 2-M0+ TYPE 3-M0+
运行模式	O	O
待机模式	O	O
深度待机模式	-	O

运行模式概述

运行模式由选择作为主控时钟的时钟确定。基本时钟通过主控时钟分频获得，向 CPU 、AHB 总线以及 APB 总线提供 CPU、总线及大部分外设运行的时钟。

源时钟频率是动态变化的。不使用主或副振荡器时，源时钟振荡器停止。

具体取决于选作为主控时钟的时钟，运行模式有以下几类：

- 高速 CR 运行模式

此模式下，高速 CR 振荡器时钟作为主控时钟。不使用主或副振荡器时，相应振荡器停止。PLL 倍频电路的状态根据 PLLE 位的设置而变化。低速 CR 振荡器总是设置为活动状态。释放复位后变为此模式。

- 主运行模式

此模式下，选择主振荡器时钟选作为主控时钟。高速 CR 振荡器、PLL 倍频电路及副振荡器的状态分别根据 HCRE 位、PLLE 位及 SOSCE 位的设置而变化。低速 CR 振荡器总是设置为活动状态。

- PLL 运行模式

此模式下，通过主振荡器时钟或高速 CR 振荡器时钟倍频获得的 PLL 时钟作为主控时钟。低速 CR 振荡器总是设置为活动状态。副振荡器的状态根据 SOSCE 位的设置而变化。高速 CR 振荡器及主振荡器的状态根据 PSW_TMR 寄存器中 PINC 位的设置而变化。

- 低速 CR 运行模式

此模式下，选用低速 CR 振荡器时钟为主控时钟。副振荡器的状态根据 SOSCE 位的设置而变化。此模式下，主振荡器、高速 CR 振荡器及 PLL 倍频电路不可用。

- 子运行模式

此模式下，副振荡器时钟作为主控时钟。低速 CR 振荡器总是设置为活动状态。此模式下，主振荡器、高速 CR 振荡器及 PLL 倍频电路不可用。

休眠模式概述

休眠模式归属于待机模式。休眠模式用于停止 CPU 时钟。在休眠模式时，CPU 停止运行以降低功耗。连接到 AHB 及 APB 总线时钟的资源则继续运行。根据休眠模式转换时的主控时钟，休眠模式分为以下几类：

- 高速 CR 休眠模式

选择高速 CR 振荡器时钟为主控时钟时，如请求转换为 SLEEP 模式，系统改变为高速 CR 休眠模式。此模式下，PLL 倍频电路、主或副振荡器的状态根据 PLLE、MOSCE 或 SOSCE 位的设置而变化。低速 CR 振荡器总是设置为活动状态。

- 主休眠模式

主时钟选为主控时钟时，如请求转换至休眠模式，系统改变为主休眠模式。此模式下，高速 CR 振荡器、PLL 倍频电路及副振荡器的状态分别根据 HCRE 位、PLLE 位及 SOSCE 位的设置而变化。低速 CR 振荡器总是设置为活动状态。

- PLL 休眠模式

主 PLL 时钟选为主控时钟时，如请求转换至休眠模式，系统改变为 PLL 休眠模式。此模式下，低速 CR 振荡器总是设置为活动状态。副振荡器的状态根据 SOSCE 位的设置而变化。高速 CR 振荡器及主振荡器的状态根据 PSW_TMR 寄存器中 PINC 位的设置而变化。

- 低速 CR 休眠模式

低速 CR 时钟选为主控时钟时，如请求转换至休眠模式，则系统改变为低速 CR 休眠模式。此模式下，副振荡器的状态根据 SOSCE 位的设置而变化。同时，主振荡器、高速 CR 振荡器及 PLL 倍频电路在此模式下不可用。

■ 副休眠模式

副时钟选为主控时钟时，如请求转换至休眠模式，则系统改变为副休眠模式。此模式下，低速 CR 振荡器总是设置为活动状态。同时，主振荡器、高速 CR 振荡器及 PLL 倍频电路在此模式下不可用。

计时器模式概述

计时器模式归属于待机模式。计时器模式用于停止提供基本时钟。计时器模式下停止 CPU 时钟、AHB 总线时钟及所有 APB 总线时钟，以降低功耗。于是，除振荡器、PLL、硬件看门狗计时器、计时计数器、RTC、时钟故障检测器及低压检测电路外，所有功能停止。

根据转换至计时器模式时的主控时钟，计时器模式有以下几类：

■ 高速 CR 计时器模式

选择高速 CR 振荡器时钟为主控时钟时，如请求转换至计时器模式，则系统改变为高速 CR 计时器模式。此模式下，PLL 倍频电路、主或副振荡器的状态根据 PLLE、MOSCE 或 SOSCE 位的设置而变化。低速 CR 振荡器总是设置为活动状态。

■ 主计时器模式

主时钟选为主控时钟时，如请求转换至计时器模式，则系统改变为主计时器模式。此模式下，高速 CR 振荡器、PLL 倍频电路及副振荡器的状态分别根据 HCRE 位、PLLE 位及 SOSCE 位的设置而变化。低速 CR 振荡器总是设置为活动状态。

■ PLL 计时器模式

选择主 PLL 时钟为主控时钟时，如请求转换至计时器模式，则系统改变为 PLL 计时器模式。此模式下，低速 CR 振荡器总是设置为活动状态。副振荡器的状态根据 SOSCE 位的设置而变化。高速 CR 振荡器及主振荡器的状态根据 PSW_TMR 寄存器中 PINC 位的设置而变化。

■ 低速 CR 计时器模式

选择低速 CR 时钟为主控时钟时，如请求转换至计时器模式，则系统改变为低速 CR 计时器模式。此模式下，副振荡器的状态根据 SOSCE 位的设置而变化。此模式下，主振荡器、高速 CR 振荡器及 PLL 倍频电路不可用。

■ 副计时器模式

副时钟选为主控时钟时，如请求转换至计时器模式，则系统变更为副计时器模式。此模式下，副振荡器及低速 CR 振荡器总是设置为活动状态。此时，主振荡器、高速 CR 振荡器及 PLL 倍频电路不可用。

RTC 模式概述

RTC 模式归属于待机模式。 RTC 模式下振荡停止（除副振荡器以外）。除计时计数器、RTC 及低压检测电路以外，所有功能停止。

停止模式概述

停止模式归属于待机模式。停止模式下停止所有振荡操作。使能此模式会停止低压检测电路之外的所有功能。

深度待机 RTC 模式概述

深度待机 RTC 模式属于深度待机模式。深度待机 RTC 模式时停止振荡（除副振荡器以外）。除 RTC 及低压检测电路之外，所有功能也停止。此模式时关闭 RTC、低压检测电路、 CPU（除 GPIO 以外）、片上闪存、片上 SRAM* 及芯片内的外设功能。

深度待机停止模式概述

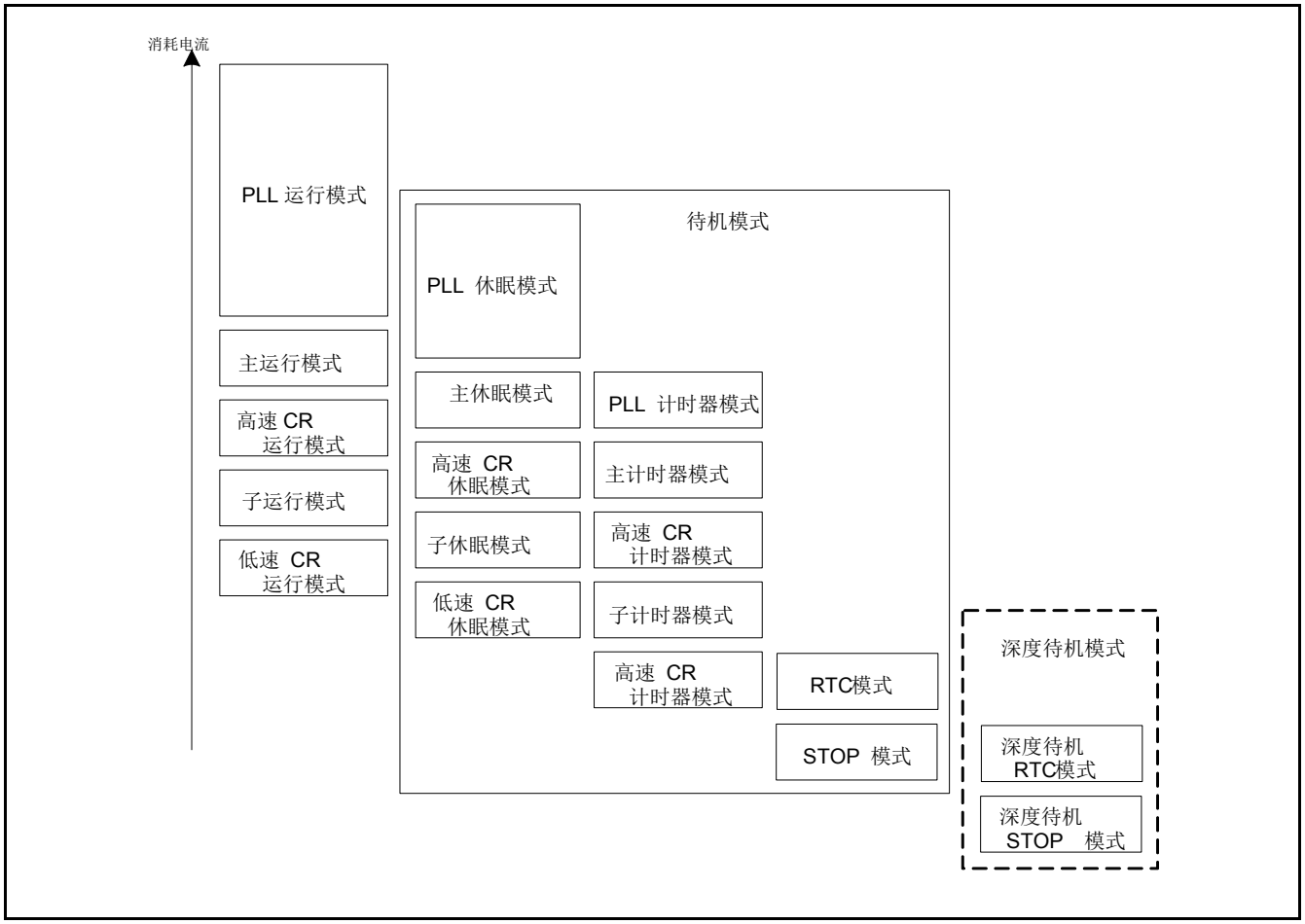
深度待机停止模式属于深度待机模式。深度待机停止模式停止所有振荡。除低压检测电路之外，所有功能停止。此模式关闭 RTC、低压检测电路、CPU （除 GPIO 以外）、片上闪存、片上 SRAM* 及芯片内的外设功能。

*: 即使在深度待机模式下也可保留片上 SRAM 中的数据。
如要保留片上 SRAM 数据，片上 SRAM 需上电。

CPU 操作模式与耗电值之间的关系。

Figure 1-1 所示为 CPU 操作模式与耗电值之间的关系。

Figure 1-1 CPU 操作模式与耗电值之间的关系



注意事项:

- *Figure 1-1 仅概述了各模式下耗电的量级关系。实际耗电值取决于各模式下的振荡器及 PLL 启动条件，或所选时钟配置的频率及其他要素。*

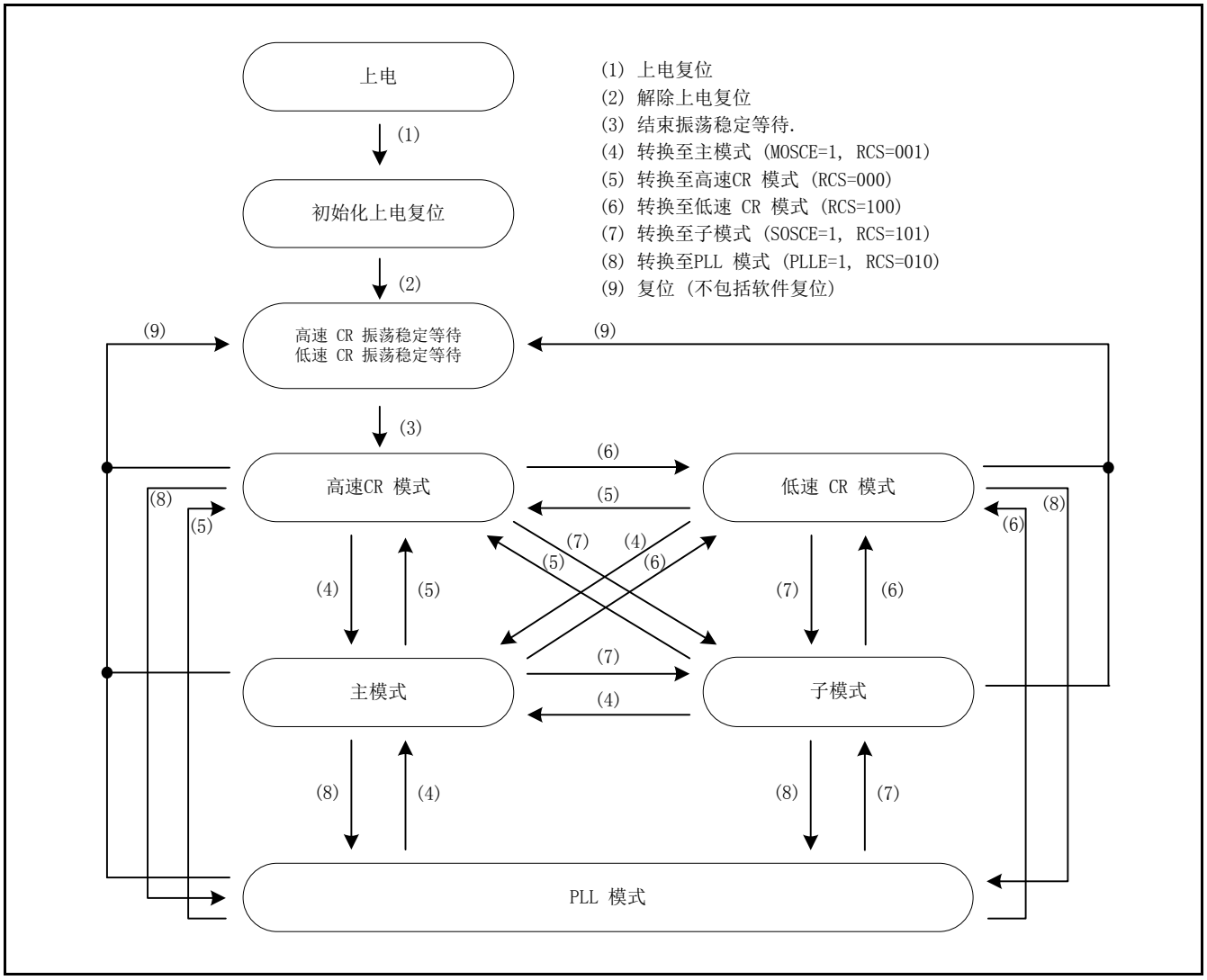
2. CPU 操作模式配置

本章说明 CPU 操作模式的配置

CPU 操作模式转换图

Figure 2-1 所示为 CPU 操作模式转换图

Figure 2-1 CPU 操作模式转换图



■ 高速 CR 模式

此模式下，高速 CR 振荡器时钟作为主控时钟。

■ 主模式

此模式下，主振荡器时钟作为主控时钟。

■ 低速 CR 模式

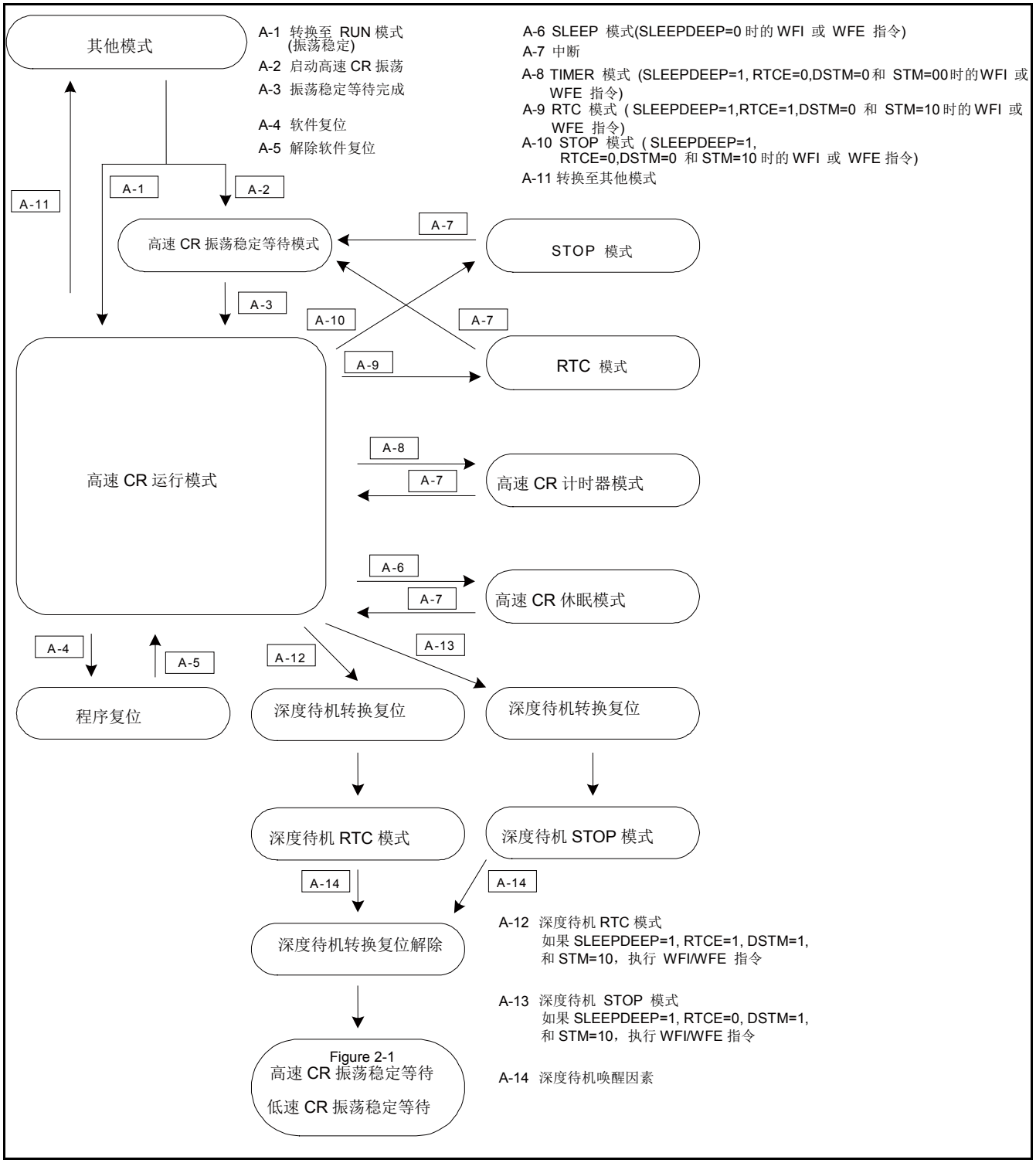
此模式下，低速 CR 振荡器时钟作为主控时钟。

- 子模式
  此模式下，副振荡器时钟作为主控时钟。
- PLL 模式
  此模式下，PLL 振荡器时钟作为主控时钟。

■ 高速 CR 模式转换图

高速 CR 模式下，高速 CR 振荡器时钟作为主控时钟。

Figure 2-2 高速 CR 模式转换图



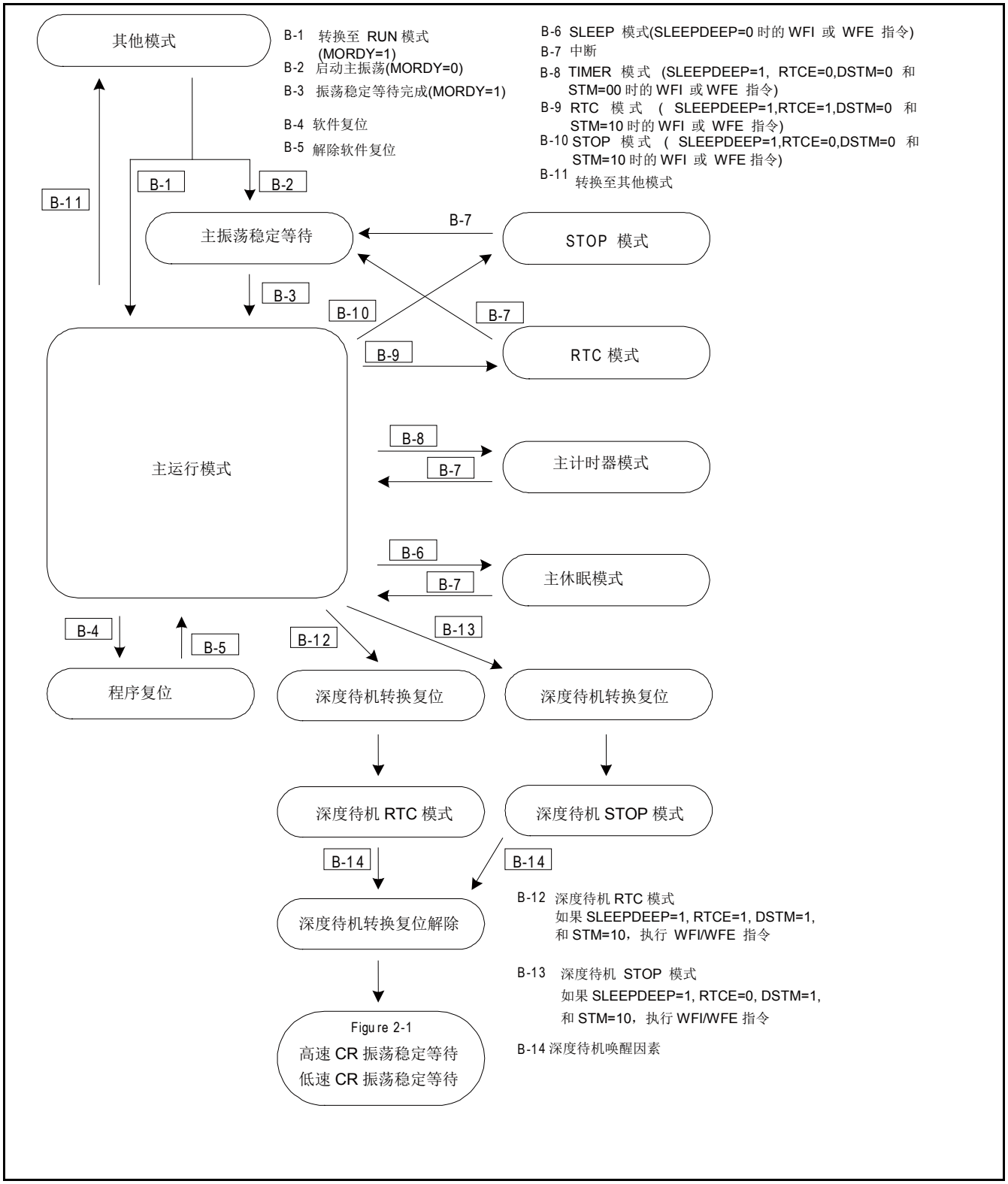
注意事项：

- *有些产品不具有深度待机 RTC 模式和深度待机停止模式。参见 Table 1-1（低功耗模式列表）。*

主模式转换图

主模式下，主振荡器时钟作为主控时钟。

Figure 2-3 主模式转换图



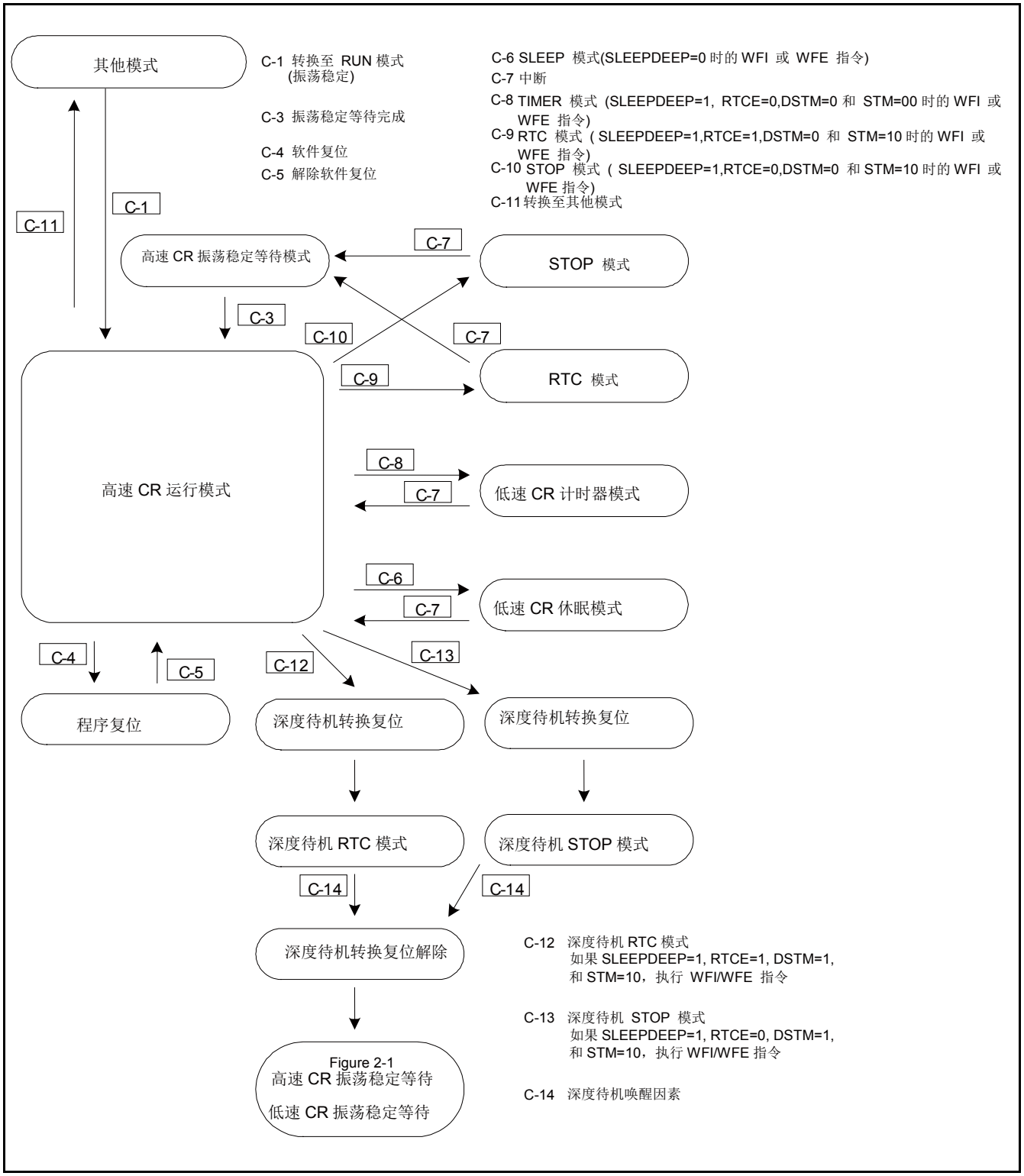
注意事项：

- *有些产品不具有深度待机 RTC 模式和深度待机停止模式。参见 Table 1-1（低功耗模式列表）。*

低速 CR 模式转换图

低速 CR 模式下，低速 CR 振荡器时钟作为主控时钟。

Figure 2-4 低速 CR 模式转换图



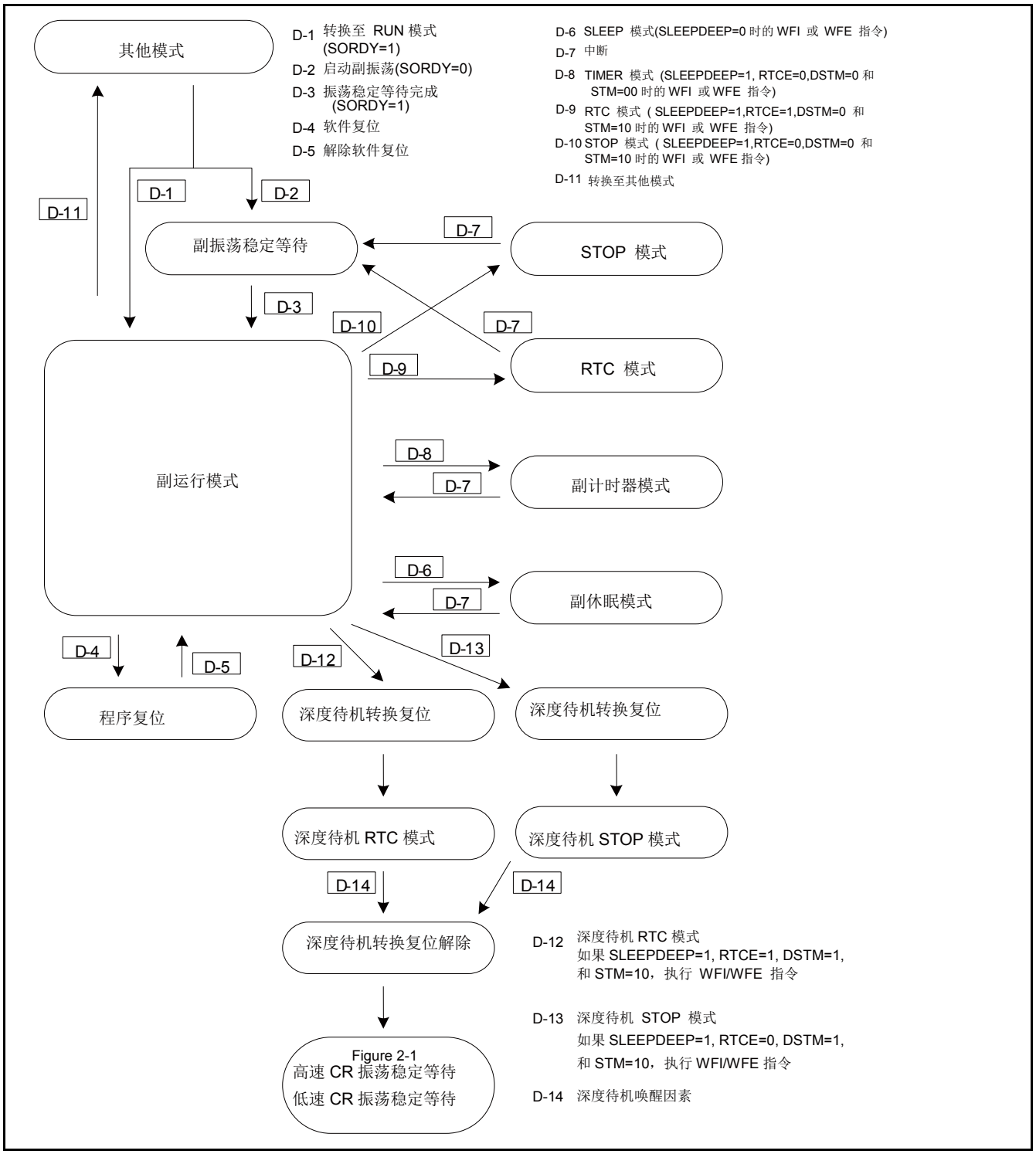
注意事项：

- *有些产品没有配备深度待机 RTC 模式以及深度待机停止模式。参见 Table 1-1（低功耗模式列表）。*

子模式转换图

子模式下，副振荡器时钟作为主控时钟。

Figure 2-5 子模式转换图



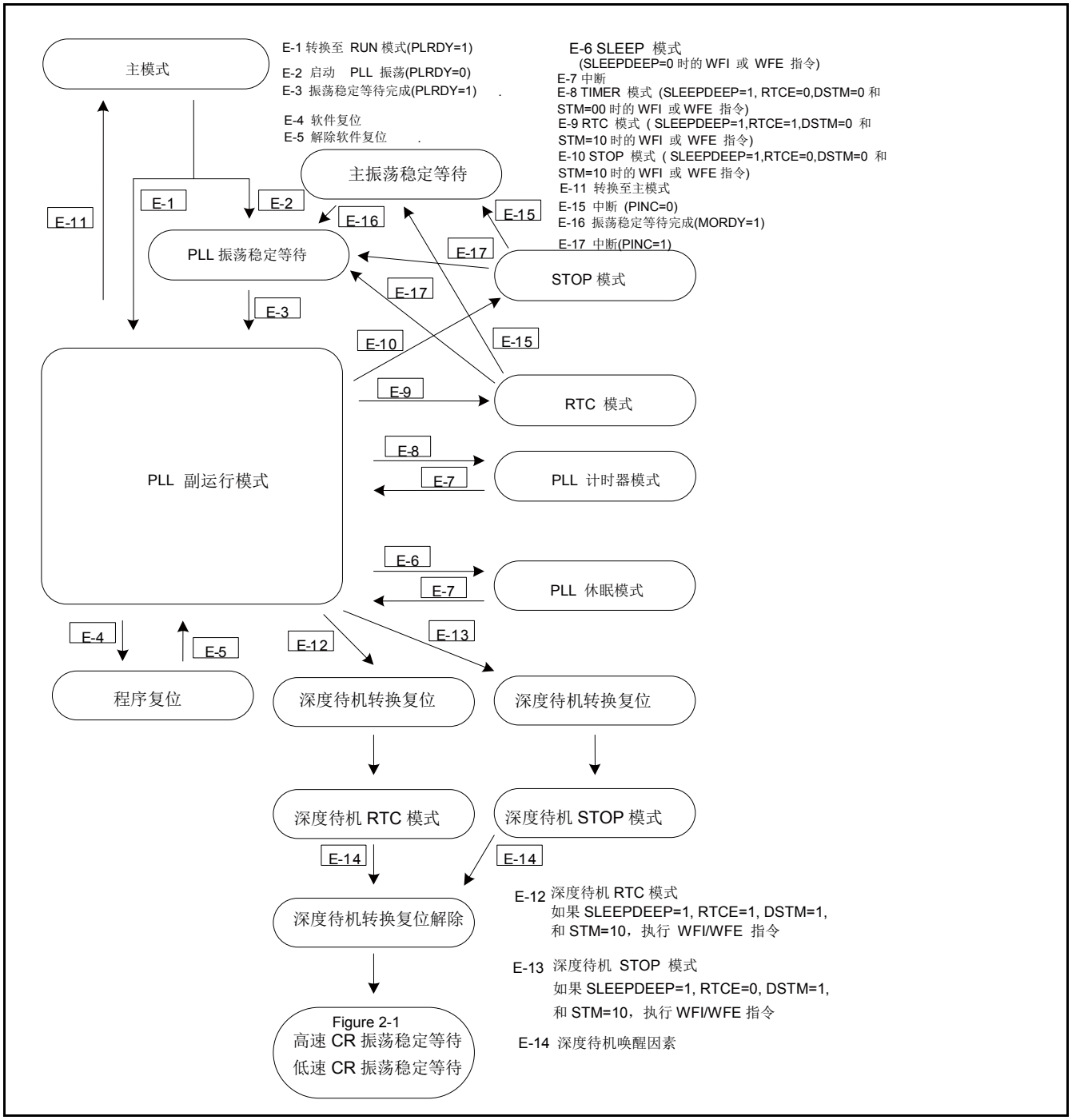
注意事项：

- *有些产品不具有深度待机 RTC 模式和深度待机停止模式。参见 Table 1-1（低功耗模式列表）。*

PLL 模式转换图

PLL 模式下，主 PLL 时钟作为主控时钟。

Figure 2-6 PLL 模式转换图



注意事项：

- *有些产品不具有深度待机 RTC 模式和深度待机停止模式。参见 Table 1-1（低功耗模式列表）。*

MOSCE ：系统时钟模式控制寄存器 (SCM_CTL) 的 MOSCE 位
SOSCE ：系统时钟模式控制寄存器 (SCM_CTL) 的 SOSCE 位
PLLE ：系统时钟模式控制寄存器 (SCM_CTL) 的 PLLE 位
RCS ：系统时钟模式控制寄存器 (SCM_CTL) 的 RCS 位
MORDY ：系统时钟模式状态寄存器 (SCM_STR) 的 MORDY 位
SORDY ：系统时钟模式状态寄存器 (SCM_STR) 的 SORDY 位
PLRDY ：系统时钟模式状态寄存器 (SCM_STR) 的 PLRDY 位
PINC ：PLL 时钟振荡等待时间设置寄存器 (PSW_TMR) 的 PINC 位
*：有关 SCM_CTL、SCM_STR 及 PSW_TMR 寄存器的信息，参见“时钟”一章。

注意事项：

- *从低速 CR 计时器模式、副计时器模式、RTC 模式、停止模式、深度待机 RTC 模式或深度待机停止模式中唤醒时，应保证内置调节器转换为操作模式的电压稳定等待时间（几百微秒）。等待时间结束后，系统从各运行模式唤醒。*

3. 待机模式的操作

本节说明待机模式的操作。
待机模式有休眠模式（高速 CR 休眠、主休眠、PLL 休眠、低速 CR 休眠及副休眠）、计时器模式（高速 CR 计时器、主计时器、PLL 计时器、低速 CR 计时器及副计时器）、RTC 模式和停止模式四种类型。

待机模式下的时钟操作状态

下表所示为休眠、计时器、RTC 以及停止模式下振荡器时钟、CPU 时钟、AHB 总线时钟及 APB 总线时钟的状态。

Table 3-1 休眠模式下的时钟操作状态

	休眠模式				
	高速 CR 休眠模式	**主休眠模式**	**PLL 休眠模式**	**低速 CR 休眠模式**	**副休眠模式**
高速 CR 时钟	操作	根据 HCRE 位、MCSVE 位及 FCSDE 位的设置而变化。	根据 PINC 位、HCRE 位、MCSVE 位及 FCSDE 位的设置而变化。	停止	停止
主时钟	根据 MOSCE 位的设置而变化。	操作	根据 PINC 位及 MOSCE 位的设置而变化。	停止	停止
主 PLL 时钟	根据 PLLE 位的设置而变化。	根据 PLLE 位的设置而变化。	操作	停止	停止
低速 CR 时钟	操作	操作	操作	操作	操作
副时钟	根据 SOSCE 位的设置而变化。	根据 SOSCE 位的设置而变化。	根据 SOSCE 位的设置而变化。	根据 SOSCE 位的设置而变化。	操作
USB PLL 时钟	根据 UPLLEN 位的设置而变化。	根据 UPLLEN 位的设置而变化。	根据 UPLLEN 位的设置而变化。	停止	停止
CPU 时钟	停止				
AHB 总线时钟	高速 CR 时钟	主时钟	PLL 时钟	低速 CR 时钟	副时钟
APB0 总线时钟	高速 CR 时钟	主时钟	PLL 时钟	低速 CR 时钟	副时钟
APB1 总线时钟	高速 CR 时钟	主时钟	PLL 时钟	低速 CR 时钟	副时钟
	*根据 APBC1EN 位的设置确定是否使能操作。				

Table 3-2 计时器模式下的时钟操作状态

	计时器模式				
	高速 CR 计时器模式	**主计时器模式**	**PLL 计时器模式**	**低速 CR 计时器模式**	**副计时器模式**
高速 CR 时钟	操作	根据 HCRE 位、MCSVE 位及 FCSDE 位的设置而变化。	根据 PINC 位、HCRE 位、MCSVE 位及 FCSDE 位的设置而变化。	停止	停止
主时钟	根据 MOSCE 位的设置而变化。	操作	根据 PINC 位及 MOSCE 位的设置而变化。	停止	停止
主 PLL 时钟	根据 PLLE 位的设置而变化。	根据 PLLE 位的设置而变化。	操作	停止	停止
低速 CR 时钟	操作	操作	操作	操作	操作
副时钟	根据 SOSCE 位的设置而变化。	根据 SOSCE 位的设置而变化。	根据 SOSCE 位的设置而变化。	根据 SOSCE 位的设置而变化。	操作
USB PLL 时钟	停止				
CPU 时钟	停止				
AHB 总线时钟	停止				
APB0 总线时钟	停止				
APB1 总线时钟	停止				

Table 3-3 RTC 模式和停止模式下的时钟操作状态

	RTC 模式	**停止模式**
高速 CR 时钟	停止	停止
主时钟		
主 PLL 时钟		
低速 CR 时钟		
副时钟	操作	
USB PLL 时钟	停止	
CPU 时钟		
AHB 总线时钟		
APB0 总线时钟		
APB1 总线时钟		

MOSCE ： 系统时钟模式控制寄存器 (SCM_CTL) 的 MOSCE 位
SOSCE ： 系统时钟模式控制寄存器 (SCM_CTL) 的 SOSCE 位
PLLE ： 系统时钟模式控制寄存器 (SCM_CTL) 的 PLLE 位
HCRE ： 系统时钟模式控制寄存器 (SCM_CTL) 的 HCRE 位
MCSVE ： CSV 控制寄存器 (CSV_CTL) 的 MCSVE 位
FCSDE ： CSV 控制寄存器 (CSV_CTL) 的 FCSDE 位

PINC ： PLL 时钟振荡等待时间设置寄存器 (PSW_TMR) 的 PINC 位
APBC1EN ： 外设总线时钟分频寄存器 (APBC1_PSR) 的 APBC1EN 位

*：有关 SCM_CTL 及 APBC1_PSR 寄存器，参见“时钟”一章。

待机模式的唤醒因素

下表所示为系统从休眠、计时器、RTC 及停止模式唤醒的因素。

Table 3-4 待机模式的唤醒因素

	休眠模式	计时器模式	RTC 模式	停止模式
复位唤醒因素	– INITX 引脚输入复位 – 低压检测复位 – 软件看门狗复位 – 硬件看门狗复位 – 时钟故障检测复位 – 异常频率检测复位	– INITX 引脚输入复位 – 低压检测复位 – 硬件看门狗复位 – 时钟故障检测复位 – 异常频率检测复位（主计时器模式、PLL 计时器模式）	– INITX 引脚输入复位 – 低压检测复位	– INITX 引脚输入复位 – 低压检测复位
中断唤醒因素	– 各外设发出的有效中断信号	– NMI 中断 – 外部中断 – I2C 从属中断 – 硬件看门狗计时器中断 – USB 唤醒中断 – 计时计数器中断 – RTC 中断 – HDMI-CEC/遥控接收中断 – 低压检测中断	– NMI 中断 – 外部中断 – I2C 从属中断 – USB 唤醒中断 – RTC 中断 – HDMI-CEC/遥控接收中断 – 低压检测中断	– NMI 中断 – 外部中断 – I2C 从属中断 – USB 唤醒中断 – 低压检测中断

3.1 休眠模式(高速 CR 休眠、主休眠、PLL 休眠、低速 CR 休眠和副休眠模式)操作

休眠模式属于待机模式，使能休眠模式停止 CPU 时钟，可降低功耗。

休眠模式的功能

■ CPU 及片上存储器

休眠模式下，停止 CPU 时钟。 AHB 总线时钟继续操作。片上存储器继续操作并保留数据。

■ 外设

休眠模式下，APB0 总线时钟仍处于活动状态。APB1 总线时钟的状态根据 APBC1EN 位的设置而变化。外设在转换时设置的状态下操作。

■ 计时计数器及 RTC

计时计数器和 RTC 不受休眠模式影响，仍然按转换至休眠模式前的设置继续操作。

■ 振荡器时钟

Table 3-1 所示为各振荡器时钟的状态。

■ 复位及中断

可通过复位和中断从计时器模式中唤醒。

引脚状态

休眠模式时保持所有引脚设置。

休眠模式设置步骤

执行以下步骤转换至休眠模式。

1. Cortex-M0+ 系统控制寄存器 SLEEPDEEP 位设置为 "0"。
2. 执行 WFI 或 WFE 指令。

根据系统时钟模式状态寄存器 (SCM_STR) 的 RCM[2:0] 位所指示的当前时钟模式，系统转换至相应的休眠模式。

有关系统时钟模式状态寄存器 (SCM_STR) ，参见“时钟”一章。

休眠模式唤醒

下列情况中，CPU 从休眠模式中唤醒：

■ 复位唤醒

如发生复位（INITX 引脚输入复位、低压检测复位、软件看门狗复位、硬件看门狗复位、时钟监视器复位或异常频率检测复位），CPU 在任何时钟模式下均改变为高速 CR 运行模式。

■ 中断唤醒

休眠模式时如收到外设的有效中断信号，CPU 从休眠模式唤醒并改变为运行模式，适合系统时钟模式状态寄存器 (SCM_STR) 的 RCM[2:0]位所指示的时钟模式。

Table 3-5 CPU 从休眠模式中断唤醒后的操作模式

	主控时钟转换为休眠模式前的状态				
	RCM=000（高速 CR 振荡器）	RCM=001（主振荡器）	RCM=010（PLL 振荡器）	RCM=100（低速 CR 振荡器）	RCM=101 （副振荡器）
中断唤醒后的操作模式	高速 CR 运行模式	主运行模式	PLL 运行模式	低速 CR 运行模式	子运行模式

RCM: 系统时钟模式状态寄存器 (SCM_STR) 的 RCM[2:0] 位

*有关 SCM_CTL 及 SCM_STR 寄存器，参见“时钟”一章。

■ 唤醒时等待振荡稳定

复位唤醒 CPU 时，需等待高速和低速 CR 时钟振荡稳定。中断唤醒 CPU 时，无需等待振荡稳定。

3.2 计时器模式(高速 CR 计时器、主计时器、PLL 计时器、低速 CR 计时器和副计时器模式)操作

计时器模式时停止提供基本时钟。这会停止 CPU 时钟、AHB 总线时钟及所有 APB 总线时钟，更大程度地降低功耗。此模式下，除振荡器、PLL、硬件看门狗计时器、计时计数器、RTC、时钟故障检测器及低压检测电路外，所有功能停止。

计时器模式的功能

■ CPU 及片上存储器

计时器模式下，向 CPU 提供的 CPU 时钟以及向片上存储器和 DMA 控制器提供的 AHB 总线时钟停止运行。但可保留片上存储器的内容。调试功能也停止。

■ 外设

计时器模式下，所有 APB 时钟停止，除硬件看门狗计时器、计时计数器、RTC、时钟监视器及低压检测电路之外的所有资源也都停止在上一状态。

■ 计时计数器及 RTC

计时计数器和 RTC 不受计时器模式的影响，仍然按转换至计时器模式前的设置继续操作。

■ 振荡器时钟

Table 3-2 所示为各振荡器时钟的状态。

■ 复位及中断

可通过复位和中断从计时器模式唤醒。

■ 外部总线

计时器模式下外部总线停止。

■ 引脚状态

系统可控制是否保持以下两种状态：外部引脚变为计时器模式之前的状态或变为高阻抗状态之前的状态，具体取决于待机模式控制寄存器 (STB_CTL) SPL 位的设置。

计时器模式设置步骤

执行以下步骤，转换至计时器模式。

1. 将 RTC 模式控制寄存器 (PMD_CTL) 的 RTCE 位设置为 "0"。
2. KEY 位写入 "0x1ACC" ，DSTM 位写入 "0" ，且待机模式控制寄存器 (STB_CTL) 的 STM 位写入 "0b00"。用 SPL 位设置计时器模式下各引脚的状态。
3. 将 Cortex-M0+ 系统控制寄存器的 SLEEPDEEP 位设置为 "1"。
4. 执行 WFI 或 WFE 指令。
   根据系统时钟模式状态寄存器 (SCM_STR) 的 RCM[2:0] 位所指示的当前时钟模式，系统转换至相应的计时器模式。

注意事项：

- *不具有深度待机 RTC 模式以及深度待机 STOP 模式的产品没有 DSTM 位。参见 Table 1-1。*

计时器模式唤醒

下列情形中，CPU 从计时器模式唤醒。

■ 复位唤醒

如发生复位（INITX 引脚输入复位、低压检测复位、硬件看门狗复位、时钟监视器复位或异常频率检测复位（主计时器模式、PLL、计时器模式）），CPU 在任何时钟模式下均改变为高速 CR 运行模式。

此模式下没有软件看门狗复位，所以不能通过此复位唤醒 CPU。

■ 中断唤醒

计时器模式时如收到有效的 NMI 中断、外部中断、硬件看门狗计时器中断、计时计数器中断、RTC 中断或低压检测中断请求，CPU 从计时器模式唤醒并改变为运行模式，适合 SCM_STR 寄存器的 RCM[2:0] 位所指示的时钟模式。

Table 3-6 CPU 从计时器模式中断唤醒后的操作模式

	主控时钟转换为计时器模式前的状态				
	RCM=000（高速 CR 振荡器）	RCM=001（主振荡器）	RCM=010（PLL 振荡器）	RCM=100（低速 CR 振荡器）	RCM=101 （副振荡器）
中断唤醒后的操作模式	高速 CR 运行模式	主运行模式	PLL 运行模式	低速 CR 运行模式	子运行模式

■ 唤醒时等待振荡稳定

复位唤醒 CPU 时，需等待高速和低速 CR 时钟振荡稳定。如果是中断唤醒 CPU ，无需等待振荡稳定。

■ 唤醒时等待内置调节器电压稳定

从低速 CR 计时器模式或副计时器模式中复位或中断唤醒时，自动保证内置调节器操作模式转换的电压稳定等待时间（几百微秒）。等待时间结束后，执行唤醒操作。

注意事项：

- *如果用于唤醒的中断优先级未达到唤醒 CPU 的级别，则此中断会唤醒时钟，而 CPU 将仍处于停止状态不被唤醒。因此，需要将中断优先级设置为能唤醒 CPU 的级别，才能唤醒 CPU。*
- *转换至计时器模式前，确保不要设置为 Table 3-4 所列从计时器模式唤醒的因素。如设置了这些中断因素，将其清除。*
- *如调试时转换至计时器模式，由于停止了向 CPU 提供时钟，不能通过 ICE 唤醒至运行模式。可采用复位或中断唤醒。*
- *转换至低速 CR 计时器模式或副计时器模式时，应在转换前终止闪存自动算法。*

3.3 RTC 模式操作

RTC 模式下振荡停止（除副振荡器以外）。除计时计数器、RTC 及低压检测电路之外，所有功能停止。

RTC 模式的功能

■ CPU 及片上存储器

RTC 模式下，向 CPU 提供的 CPU 时钟以及向片上存储器和 DMA 控制器提供的 AHB 总线时钟停止运行。但可保留片上存储器的内容。调试功能也停止。

■ 外设功能

RTC 模式下，所有 APB 总线时钟停止，除计时计数器、RTC、时钟监视器及低压检测电路之外的所有资源也都停止在上一状态。

■ 计时计数器及 RTC

计时计数器不受 RTC 模式的影响，按转换至 RTC 模式前的设置继续操作，且不被计时计数器中断从 RTC 模式唤醒。

RTC 不受 RTC 模式的影响，按转换至 RTC 模式前的设置继续操作。

■ 振荡时钟

Table 3-3 所示为各振荡时钟的状态。

■ 复位及中断

可通过复位和中断从 RTC 模式唤醒。

■ 外部总线

RTC 模式下外部总线停止。

■ 引脚状态

系统可控制是否保持以下两种状态：外部引脚变为计时器模式之前的状态或变为高阻抗状态之前的状态，具体取决于待机模式控制寄存器 (STB_CTL) SPL 位的设置。

RTC 模式设置步骤

执行以下步骤，以转换至 RTC 模式。

1. 系统时钟模式状态寄存器 (SCM_STR) 的 SORDY 位为 "1" 时，将 RTC 模式控制寄存器 (PMD_CTL) 的 RTCE 位设置为 "1"。
2. KEY 位写入 "0x1ACC"，DSTM 位写 "0"，且待机模式控制寄存器 (STB_CTL) 的 STM 位写入 "0b10"。用 SPL 位设置 RTC 模式下各引脚的状态。
3. Cortex-M0+ 系统控制寄存器的 SLEEPDEEP 位设置为 "1"。
4. 执行 WFI 或 WFE 指令。

注意事项：

- *不具有深度待机 RTC 模式和深度待机停止模式的产品不具有 DSTM 位。参见 Table 1-1。*

RTC 模式唤醒

以下情形中，CPU 从 RTC 模式中唤醒：

■ 复位唤醒

如发生复位（INITX 引脚输入复位、低压检测复位），CPU 在任何时钟模式下均改变为高速 CR 运行模式。这种模式下没有软件看门狗复位、硬件看门狗复位、时钟监视器复位及异常频率检测复位，所以不能通过这些复位唤醒 CPU。

■ 中断唤醒

RTC 模式时如收到有效的 NMI 中断、外部中断、RTC 中断或低压检测中断请求，CPU 从 RTC 模式唤醒并改变为运行模式，以适应 SCM_STR 寄存器的 RCM[2:0] 位所指示的时钟模式。

Table 3-7 CPU 从 RTC 模式中断唤醒后的操作模式

	主控时钟转换为 RTC 模式前的状态				
	RCM=000（高速 CR 振荡器）	RCM = 001（主振荡器）	RCM = 010（PLL 振荡器）	RCM = 100（低速 CR 振荡器）	RCM = 101（副振荡器）
中断唤醒后的操作模式	高速 CR 运行模式	主运行模式	PLL 运行模式	低速 CR 运行模式	子运行模式

■ 唤醒时等待振荡稳定

复位唤醒 CPU 时，需等待高速和低速 CR 时钟振荡稳定。如 Table 3-8 所示，如中断唤醒 CPU，转换为 RTC 模式前主控时钟可改变振荡稳定等待时间。

Table 3-8 RTC 模式中断唤醒时等待振荡稳定

		主控时钟转换为 RTC 模式前的状态				
		RCM=000（高速 CR 振荡器）	RCM = 001（主振荡器）	RCM = 010（PLL 振荡器）	RCM = 100（低速 CR 振荡器）	RCM = 101（副振荡器）
中断唤醒后等待振荡稳定	高速 CR 时钟	开启	HCR="使能"：开启 *1 其它：关闭	PINC="1" 或 HCR="使能"：开启 *1 其它：关闭	关闭	关闭
	主时钟	MOSCE="0"：关闭 MOSCE="1"：开启	开启	PINC="0" 或 MOSCE="1"：开启 其它：关闭	关闭	关闭
	主 PLL 时钟	PLLE="0"：关闭 PLLE="1"：开启	PLLE="0"：关闭 PLLE="1"：开启	开启	关闭	关闭 F
	低速 CR 时钟	开启	开启	开启	开启	开启
	副时钟	关闭*2	关闭*2	关闭*2	关闭*2	关闭*2

*1：HCRE ="1" 或 MCSVE="1" 或 FCSDE="1" 时使能 HCR。

*2：TYPE 1-M0+/TYPE 2-M0+/TYPE 3-M0+ 产品有振荡稳定等待时间，但实际上振荡会持续。通过 CSW_TMR:SOWT 位可设置最短等待时间为 "1100"。

■ 唤醒时等待内置调节器电压稳定

从 RTC 模式唤醒时，可自动保证内置调节器操作模式转换的电压稳定等待时间（几百微秒）。等待时间结束后，执行唤醒操作。

注意事项：

- *如果用于唤醒的中断优先级未达到唤醒 CPU 的级别，则此中断会唤醒时钟，而 CPU 将仍处于停止状态不被唤醒。因此，需要将中断优先级设置为能唤醒 CPU 的级别，才能唤醒 CPU。*
- *转换至 RTC 模式前，确保不要设置为 Table 3-4 所列从计时器模式唤醒的因素。如设置了这些唤醒因素，将其清除。*
- *如调试时转换至 RTC 模式，由于停止了向 CPU 提供时钟，不能通过 ICE 唤醒至运行模式。可采用复位或中断唤醒。*
- *转换至 RTC 模式时，应在转换前终止闪存自动算法。*

3.4 停止模式操作

停止模式用于停止所有振荡操作。使能此模式可停止低压检测电路以外的所有功能。

停止模式的功能

■ CPU 及片上存储器

停止模式时，向 CPU 提供的 CPU 时钟以及向片上存储器和 DMA 控制器提供的 AHB 总线时钟停止运行。但可保留片上存储器的内容。调试功能停止。

■ 外设功能

所有的 APB 总线时钟停止，且除低压检测电路之外的所有资源也都停止在上一状态。

■ 振荡器时钟

所有振荡器时钟停止。

■ 复位及中断

可通过复位和中断从停止模式中唤醒。

■ 外部总线

停止模式下外部总线停止。

■ 引脚状态

系统可控制是否保持以下两种状态：外部引脚变为计时器模式之前的状态或变为高阻抗状态之前的状态，具体取决于待机模式控制寄存器 (STB_CTL) SPL 位的设置。

停止模式设置步骤

执行以下步骤，以转换至停止模式。

1. 将 RTC 模式控制寄存器 (PMD_CTL) 的 RTCE 位设置为 "0"。
2. KEY 位写入 "0x1ACC" ，DSTM 位写入 "0"且在待机模式控制寄存器 (STB_CTL) 的 STM 位写入 "0b10" 。用 SPL 位设置停止模式下各引脚状态。
3. 将 Cortex-M0+ 系统控制寄存器的 SLEEPDEEP 位设置为 "1"。
4. 执行 WFI 或 WFE 指令。

注意事项：

- *不具有深度待机 RTC 模式和深度待机停止模式的产品不具有 DSTM 位。参见 Table 1-1。*

停止模式唤醒

以下情形中，CPU 从停止模式中唤醒：

■ 复位唤醒

如发生复位（INITX 引脚输入复位或低压检测复位），CPU 在任何时钟模式下均改变为高速 CR 运行模式。

此模式时没有软件看门狗复位、硬件看门狗复位、时钟监视器复位及异常频率检测复位，所以不能通过这些复位唤醒 CPU。

■ 中断唤醒

停止模式时如收到有效的 NMI 中断、外部中断、RTC 中断或低压检测中断请求，CPU 从停止模式唤醒并改变为运行模式，适合 SCM_STR 寄存器的 RCM[2:0] 位所指示的时钟模式。

Table 3-9 CPU 从停止模式中断唤醒后的操作模式

	主控时钟转换为停止模式前的状态				
	RCM=000（高速 CR 振荡器）	RCM=001（主振荡器）	RCM=010（PLL 振荡器）	RCM=100（低速 CR 振荡器）	RCM=101（副振荡器）
中断唤醒后的操作模式	高速 CR 运行模式	主运行模式	PLL 运行模式	低速 CR 运行模式	子运行模式

■ 唤醒时等待振荡稳定

复位唤醒 CPU 时，需等待高速和低速 CR 时钟振荡稳定。如 Table 3-10 所示，如 CPU 经中断唤醒，振荡稳定等待状态根据 CPU 变为停止模式前输出的主控时钟而变化。

Table 3-10 停止模式中断唤醒时等待振荡稳定

		主控时钟转换为停止模式前的状态				
		RCM=000（高速 CR 振荡器）	RCM=001（主振荡器）	RCM=010（PLL 振荡器）	RCM=100（低速 CR 振荡器）	RCM=101 （副振荡器）
中断唤醒后等待振荡稳定	高速 CR 时钟	开启	HCR="使能"：开启 * 其它：关闭	PINC="1" 或 HCR="使能"：开启 * 其它：关闭	关闭	关闭
	主时钟	MOSCE="0"：关闭 MOSCE="1"：开启	开启	PINC="0" 或 MOSCE="1"：开启 其它：关闭	关闭	关闭
	主 PLL 时钟	PLLE="0"：关闭 PLLE="1"：开启	PLLE="0"：关闭 PLLE="1"：开启	开启	关闭	关闭
	低速 CR 时钟	开启	开启	开启	开启	开启
	副时钟	SOSCE="0"：关闭 SOSCE="1"：开启	SOSCE="0"：关闭 SOSCE="1"：开启	SOSCE="0"：关闭 SOSCE="1"：开启	SOSCE="0"：关闭 SOSCE="1"：开启	开启

*：HCRE ="1" 或 MCSVE="1" 或 FCSDE="1" 时使能 HCR。

■ 唤醒时等待内置调节器电压稳定

CPU 从停止模式唤醒时，可自动保证内置调节器操作模式转换的电压稳定等待时间（几百微秒）。等待时间结束后，执行唤醒操作。

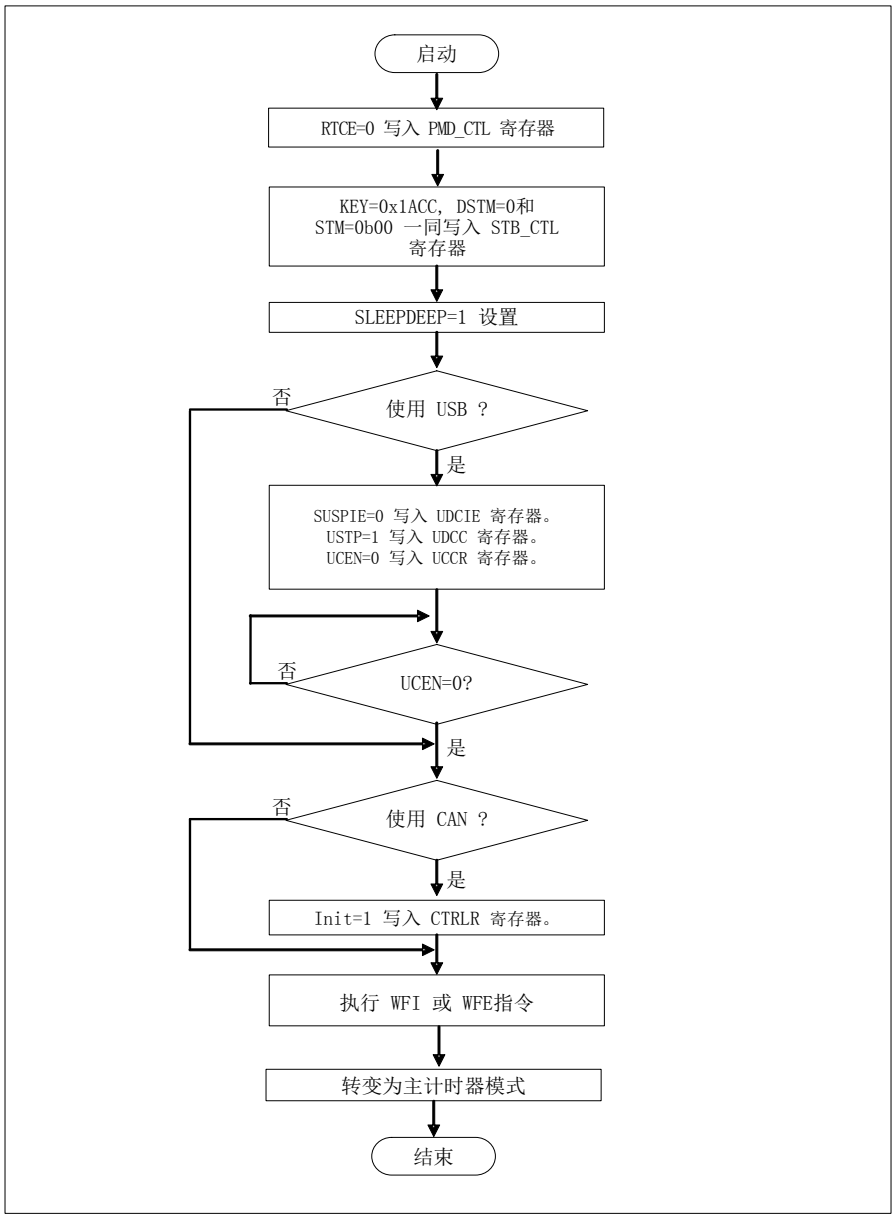
注意事项：

- *如果用于唤醒的中断优先级未达到唤醒 CPU 的级别，则此中断会唤醒时钟，而 CPU 将仍处于停止状态不被唤醒。因此，需要将中断优先级设置为能唤醒 CPU 的级别，才能唤醒 CPU。*
- *转换至停止模式前，确保不要设置为 Table 3-4 所列从计时器模式唤醒的因素。如设置了这些中断因素，将其清除。*
- *如调试时转换至停止模式，由于停止了向 CPU 提供时钟，不能通过 ICE 唤醒至运行模式。可采用复位或中断唤醒。*
- *转换至停止模式时，应在转换前终止闪存自动算法。*

4. 待机模式设置步骤示例

本节说明待机模式设置步骤示例。

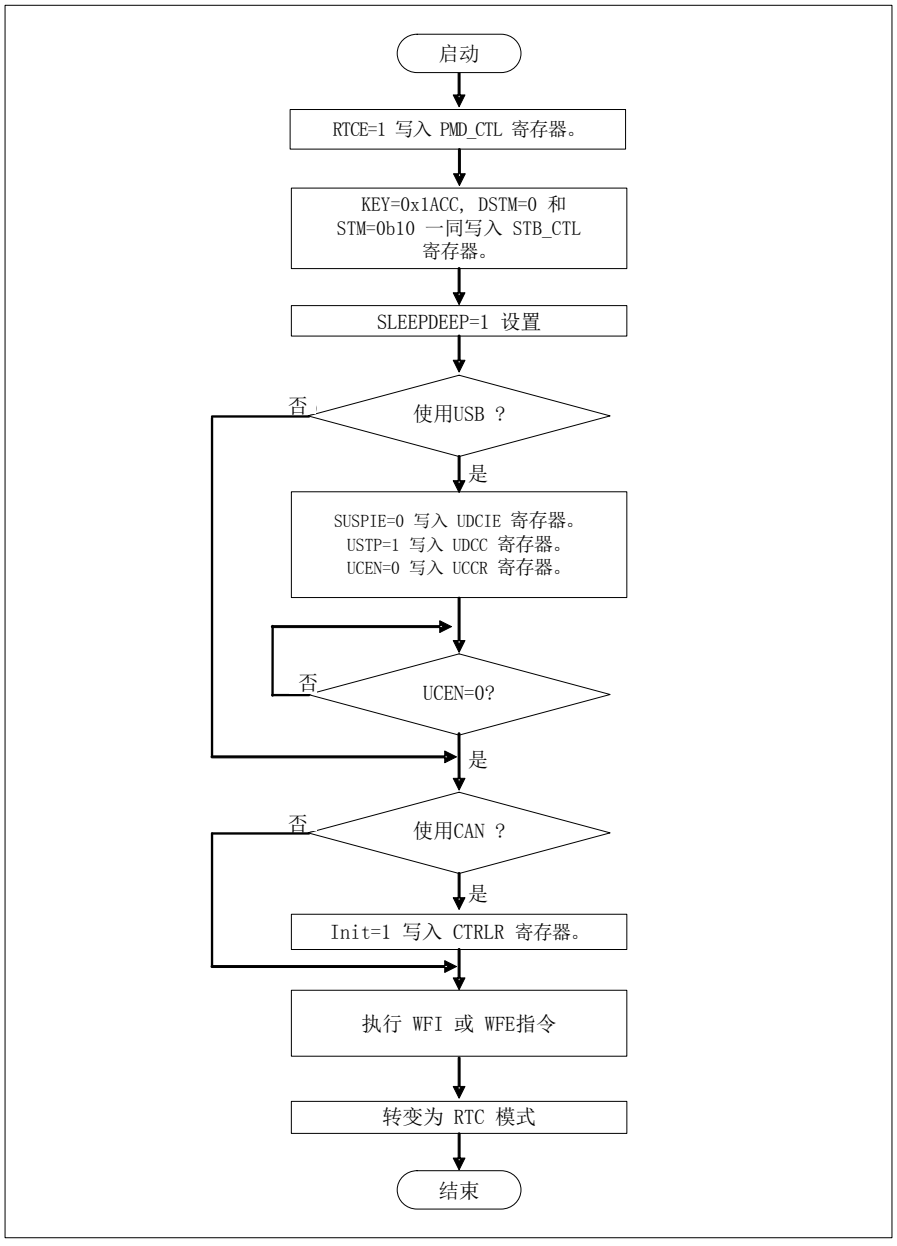
Figure 4-1 主计时器模式设置步骤示例



注意事项：

- *不具有深度待机 RTC 模式和深度待机停止模式的产品不具有 DSTM 位。参见 Table 1-1。*

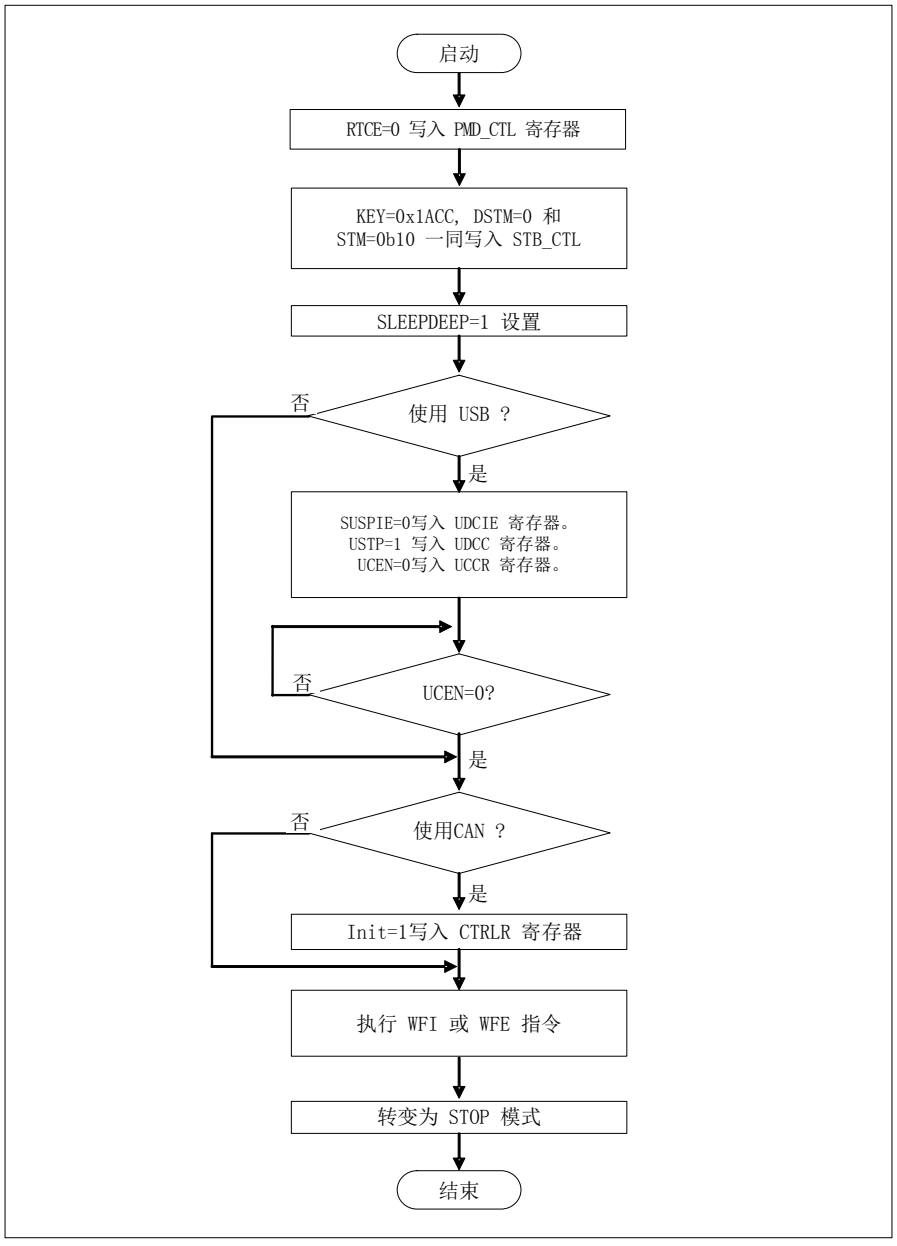
Figure 4-2 RTC 模式设置步骤示例（主时钟选为主控时钟）



注意事项：

- *转换至 RTC 模式时，应在转换前终止闪存自动算法。*
- *只有系统时钟模式状态寄存器 (SCM_STR) 的 SORDY 位为 "1" 时，将 "1" 写入 RTC 模式控制寄存器 (PMD_CTL) 的 RTCE 位才有效。*
- *不具有深度待机 RTC 模式和深度待机停止模式的产品不具有 DSTM 位。参见 Table 1-1。*

Figure 4-3 停止模式设置步骤示例（主时钟选为主控时钟）



注意事项：

- *转换至停止模式时，应在转换前终止闪存自动算法。*
- *不具有深度待机 RTC 模式和深度待机停止模式的产品不具有 DSTM 位。参见 Table 1-1。*

5. 深度待机模式操作说明

本节说明深度待机模式操作。
深度待机模式包括深度待机 RTC 模式及深度待机停止模式。

5.1 深度待机模式操作

5.1.1 深度待机模式的时钟操作状态

下表所示为深度待机 RTC 模式及深度待机停止模式时的振荡时钟、CPU 时钟、AHB 总线时钟和 APB 总线时钟的状态。

Table 5-1 深度待机模式下的时钟操作状态

	深度待机 RTC 模式	深度待机停止模式
高速 CR 时钟	停止	停止
主时钟		
主 PLL 时钟		
低速 CR 时钟		
副时钟	操作	
USB PLL 时钟	停止	
CPU 时钟		
AHB 总线时钟		
APB0 总线时钟		
APB1 总线时钟		

5.1.2 深度待机模式的唤醒因素

下表所示为深度待机 RTC 模式和深度待机停止模式的唤醒因素。

Table 5-2 深度待机模式的唤醒因素

	深度待机 RTC 模式	深度待机停止模式
深度待机唤醒因素	– INITX 引脚输入复位 – 低压检测复位	– INITX 引脚输入复位 – 低压检测复位
	– 低压检测中断 – RTC 中断 – HDMI-CEC/遥控接收中断	– 低压检测中断
	– WKUP 引脚输入	– WKUP 引脚输入

注意事项：

– *尽管从深度待机模式唤醒后仍然保留各中断因素，由于 NVIC 被深度待机转换复位初始化，系统不会执行中断进程。*

5.1.3 深度待机模式的内部电源状态及复位状态

下表所示为深度待机模式下各功能的电源状态以及深度待机转换复位的初始化状态。

Table 5-3 深度待机模式的内部电源状态及初始化状态

	产品类型	电源状态	复位状态
CPU	TYPE 2-M0+	掉电	初始化
片上闪存		掉电	*1
片上 SRAM		掉电*2	*3
RTC		上电	不初始化
HDMI-CEC/遥控接收		上电	不初始化
低压检测电路		上电	不初始化
GPIO		上电	部分初始化*4
深度待机控制模块		上电	不初始化
其他外设功能		掉电	初始化
CPU	TYPE 3-M0+	掉电	初始化
片上闪存		掉电	*1
片上 SRAM		掉电*2	*3
RTC		上电	不初始化
HDMI-CEC/遥控接收		上电	不初始化
低压检测电路		上电	不初始化
GPIO		掉电	初始化
深度待机控制模块		上电	不初始化
其他外设功能		掉电	初始化

*1：可保留片上闪存的内容。

*2：可保留片上 SRAM 的内容。
如要保留片上 SRAM 数据时，片上 SRAM 需上电。

*3：掉电时不能保留片上 SRAM 的内容。
设置保留片上 SRAM 数据，以保留片上 SRAM 数据。

*4：除 bit4:0 及 PFR0 的 CEC 之外，PFRx 寄存器初始化，其他寄存器不初始化。

5.2 深度待机 RTC 模式的操作

深度待机 RTC 模式停止振荡（除副振荡器振荡以外）。除 RTC、HDMI-CEC/遥控接收及低压检测电路之外，所有功能停止。此模式关闭 RTC、HDMI-CEC/遥控接收、低压检测电路、CPU （GPIO 除外）、片上闪存、片上 SRAM* 及芯片内的外设功能。

深度待机 RTC 模式的功能

■ CPU 及片上存储器

深度待机 RTC 模式时，向 CPU 提供的 CPU 时钟以及向片上存储器和 DMA 控制器提供的 AHB 总线时钟停止运行。此模式关闭 CPU、片上闪存及片上 SRAM*。不保留 CPU 寄存器及片上 SRAM 的内容*。保留片上闪存的内容。调试功能也停止。

*: 可保留片上 SRAM 的数据。
如要保留片上 SRAM 数据时，片上 SRAM 需上电。

■ 外设功能
所有 APB 总线时钟停止，且关闭 RTC、HDMI-CEC/遥控接收、低压检测电路及除 GPIO 之外的所有资源。

■ RTC 和 HDMI-CEC/遥控接收
RTC 和 HDMI-CEC/遥控接收不受深度待机 RTC 模式的影响，按转换至深度待机 RTC 模式前的设置继续操作。

■ 振荡时钟
Table 5-1 所示为各振荡时钟的状态。

■ 复位、中断及 WKUP 引脚输入
复位、中断及 WKUP 引脚输入可用于深度待机 RTC 模式唤醒。

■ 引脚状态
深度待机 RTC 模式下，系统可通过待机模式控制寄存器 (STB_CTL) 的 SPL 位控制外部引脚切换为 GPIO 状态或切换为高阻抗状态。

深度待机 RTC 模式的设置步骤

执行以下步骤，以转换至深度待机 RTC 模式。

1. 系统时钟模式状态寄存器 (SCM_STR) 的 SORDY 位为 "1" 时，将 RTC 模式控制寄存器 (PMD_CTL) 的 RTCE 位设置为 "1"。
2. KEY 位写入 "0x1ACC" ，DSTM 位写 "1" ，且待机模式控制寄存器 (STB_CTL) 的 STM 位写入 "0b10"。用 SPL 位设置深度待机 RTC 模式下各引脚的状态。
3. 将 Cortex-M0+ 系统控制寄存器的 SLEEPDEEP 位设置为 "1"。
4. 执行 WFI 或 WFE 指令。

注意事项：

- *不具有深度待机 RTC 模式和深度待机停止模式的产品不具有 DSTM 位。参见 Table 1-1。*

深度待机 RTC 模式的唤醒

以下情形中，CPU 从深度待机 RTC 模式唤醒。

■ 通过复位、中断及 WKUP 引脚输入唤醒

深度待机 RTC 模式时如发生复位（INITX 引脚输入复位及低压检测复位），或收到有效的 RTC 中断、HDMI-CEC/遥控接收中断、低压检测中断及 WKUP 引脚输入请求，无论是何种时钟模式，CPU 都能通过深度待机转换复位从深度待机 RTC 模式唤醒并转换为高速 CR 运行模式。

此模式是没有软件看门狗复位、硬件看门狗复位、时钟监视器复位及异常频率检测复位，所以不能通过这些复位唤醒 CPU。

■ 唤醒时等待振荡稳定

无论是什么唤醒因素，高速 CR 时钟和低速 CR 时钟都需等待振荡稳定。

■ 唤醒时等待内置调节器电压稳定

深度待机 RTC 模式唤醒时可自动保证内置调节器操作模式转换的电压稳定等待时间（几百微秒)。等待时间结束后，进行唤醒操作。

注意事项：

- *转换至深度待机 RTC 模式前，确保未设置 Table 5-2 所列深度待机 RTC 模式的唤醒因素。如设置了此类唤醒因素，将其清除。*
- *如调试期间转换至深度待机 RTC 模式，由于关闭了调试功能，不能通过 ICE 唤醒至运行模式。可通过复位、中断或 WKUP 引脚输入唤醒。*
- *转换至深度待机 RTC 模式时，应在转换前终止闪存自动算法。*

5.3 深度待机停止模式操作

深度待机停止模式停止所有振荡。除低压检测电路之外，所有功能停止。此模式关闭 RTC、HDMI-CEC/遥控接收、低压检测电路、CPU （GPIO 除外）、片上闪存、片上 SRAM* 及芯片内的外设功能。

深度待机停止模式的功能

■ CPU 及片上存储器

深度待机停止模式下，向 CPU 提供的 CPU 时钟以及向片上存储器和 DMA 控制器提供的 AHB 总线时钟停止运行。此模式关闭 CPU、片上闪存及片上 SRAM*。不保留 CPU 寄存器及片上 SRAM 的内容*。保留片上闪存的内容。调试功能也停止。

*: 可保留片上 SRAM 的数据。
如要保留片上 SRAM 数据时，片上 SRAM 需上电。

■ 外设

所有 APB 总线时钟停止，且关闭 RTC、HDMI-CEC/遥控接收、低压检测电路及除 GPIO 之外的所有资源。

■ 振荡时钟

所有振荡停止。

■ 复位及 WKUP 引脚输入

可使用复位及 WKUP 引脚输入从深度待机停止模式唤醒。

■ 引脚状态

系统可通过待机模式控制寄存器(STB_CTL) 的 SPL 位设置控制外部引脚转换为深度待机停止模式的 GPIO 状态或转换为高阻抗状态。

深度待机停止模式的设置步骤

执行以下步骤，以转换至深度待机停止模式。

1. 将 RTC 模式控制寄存器 (PMD_CTL) 的 RTCE 位设置为 "0"。
2. KEY 位写入 "0x1ACC" ，DSTM 位写 "1" ，且待机模式控制寄存器 (STB_CTL) 的 STM 位写入 "0b10"。用 SPL 位设置深度待机停止模式下各引脚的状态。
3. 将 Cortex-M0+ 系统控制寄存器的 SLEEPDEEP 位设置为 "1"。
4. 执行 WFI 或 WFE 指令。

注意事项：

- *不具有深度待机 RTC 模式和深度待机停止模式的产品不具有 DSTM 位。参见 Table 1-1。*

深度待机停止模式的唤醒

以下情形中，CPU 从深度待机停止模式中唤醒：

■ 复位、中断及 WKUP 引脚输入唤醒

深度待机停止模式时如发生复位（INITX 引脚输入复位及低压检测复位），或收到有效的低压检测中断或 WKUP 引脚输入请求，无论是何种时钟模式，CPU 都能通过深度待机转换复位从深度待机停止模式中唤醒并转换为高速 CR 运行模式。

这种模式下没有软件看门狗复位、硬件看门狗复位、时钟监视器复位及异常频率检测复位，所以不能通过这些复位唤醒 CPU。

■ 唤醒时等待振荡稳定

无论是什么唤醒因素，高速 CR 时钟和低速 CR 时钟都需等待振荡稳定。

■ 唤醒时等待内置调节器电压稳定

深度待机 STOP 模式唤醒时可自动保证内置调节器操作模式转换的电压稳定等待时间（几百微秒）。等待时间结束后，进行唤醒操作。

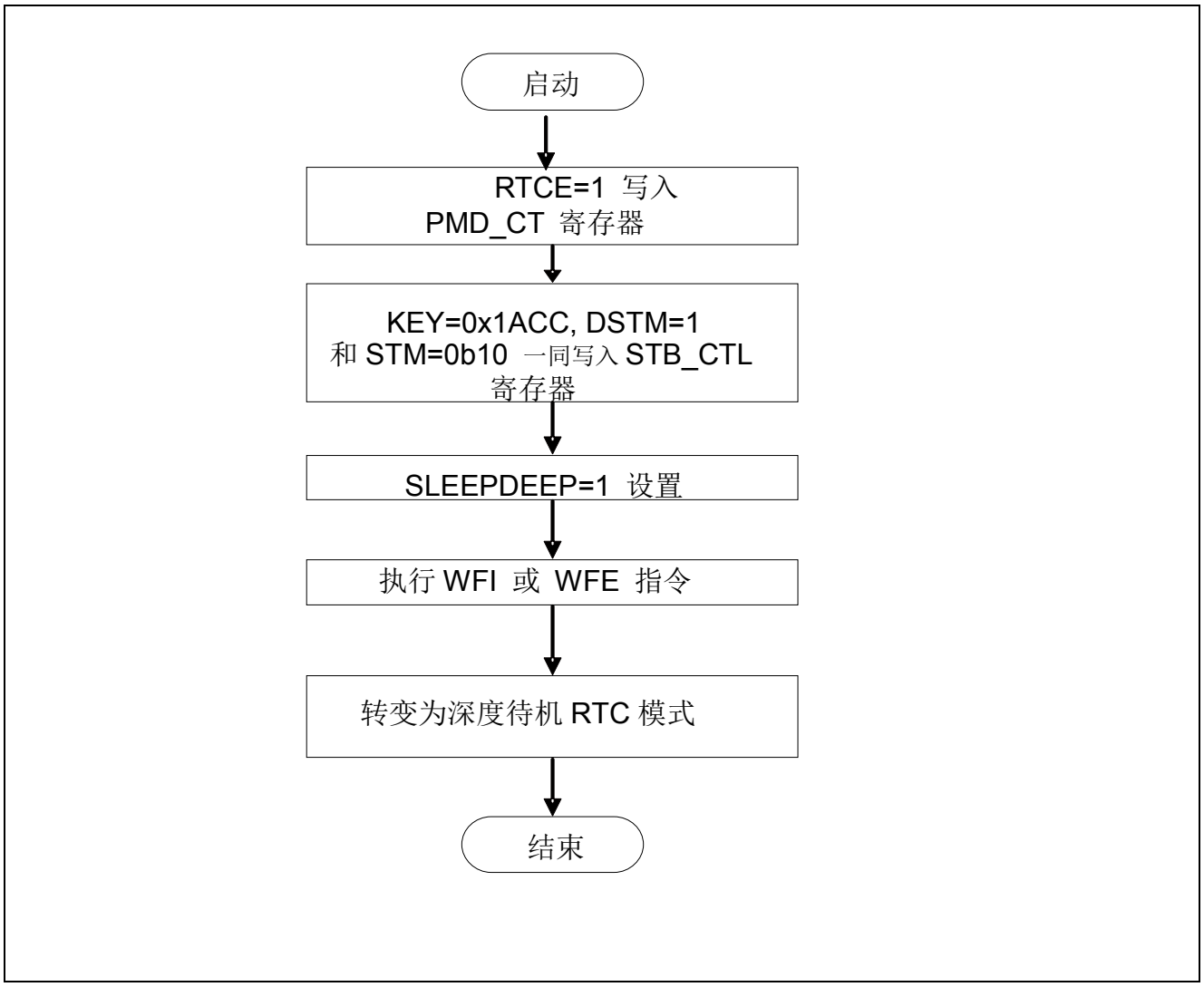
注意事项：

- *转换至深度待机 RTC 模式前，确保未设置 Table 5-2 所述的深度待机 RTC 模式的唤醒因素。如设置了此类唤醒因素，将其清除。*
- *如调试期间转换至深度待机停止模式，由于关闭了调试功能，不能通过 ICE 唤醒至运行模式。可通过复位、中断或 WKUP 引脚输入唤醒。*
- *转换至深度待机停止模式时，应在转换前终止闪存自动算法。*

6. 深度待机模式设置步骤示例

本节说明深度待机模式设置步骤示例。

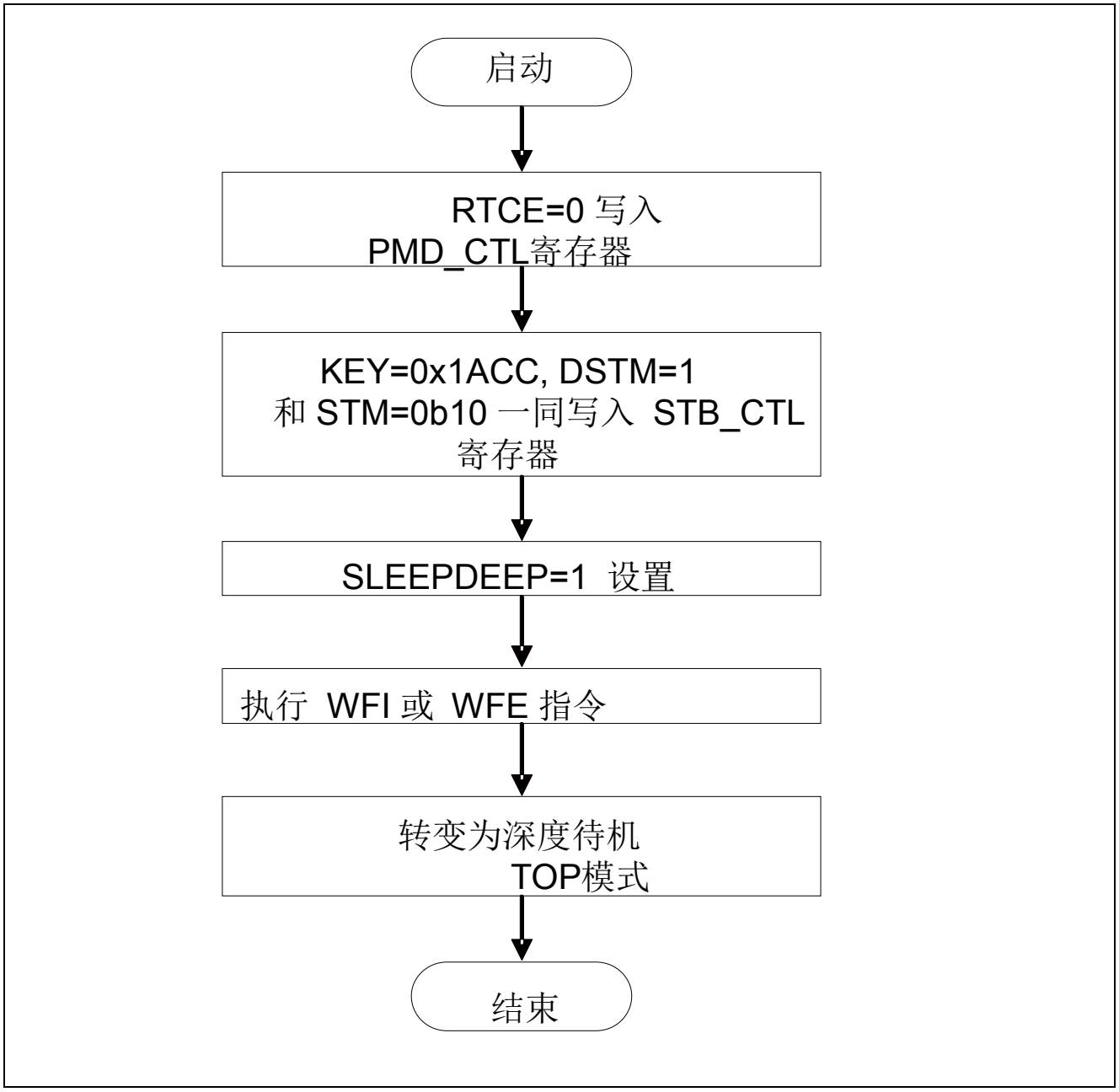
Figure 6-1 深度待机 RTC 模式设置步骤示例



注意事项：

- *转换至深度待机 RTC 模式时，应在转换前终止闪存自动算法。*
- *只有系统时钟模式状态寄存器 (SCM_STR) 的 SORDY 位为 "1" 时，将 "1" 写入 RTC 模式控制寄存器 (PMD_CTL) 的 RTCE 位才有效。*

Figure 6-2 深度待机停止模式的设置步骤示例



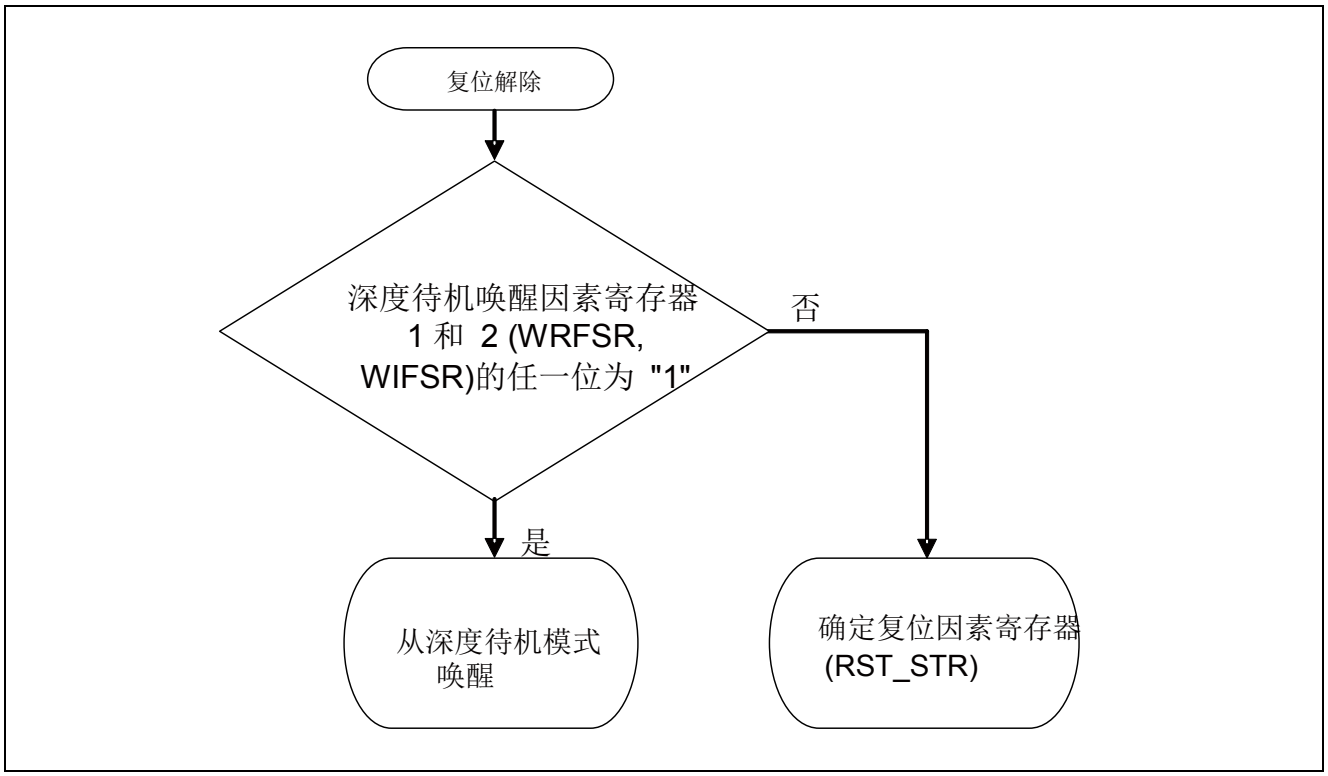
注意事项：

- *转换至深度待机 STOP 模式时，应在转换前终止闪存自动算法。*

7. 深度待机唤醒因素确定步骤

Figure 7-1 所示为深度待机模式唤醒的确定步骤示例。

Figure 7-1 深度待机唤醒因素确定步骤示例



注意事项：

- *转换至深度待机模式时，深度待机转换复位后 CPU 掉电。因此，从深度待机模式唤醒时复位因素寄存器 (RST_STR) 的值无效。*

8. 寄存器

本节说明低功耗模式所使用的寄存器。

低功耗模式寄存器列表

Table 8-1 低功耗模式寄存器列表

缩写	寄存器名称	参考章节
STB_CTL	待机模式控制寄存器	8.1

■ 深度待机控制块寄存器列表

Table 8-2 深度待机控制块寄存器列表

缩写	寄存器名称	参考章节
REG_CTL	副振荡电路电源控制寄存器	8.2
RCK_CTL	副时钟控制寄存器	8.3
PMD_CTL	RTC 模式控制寄存器	8.4
WRFSR	深度待机唤醒因素寄存器 1	8.5
WIFSR	深度待机唤醒因素寄存器 2	8.6
WIER	深度待机唤醒使能寄存器	8.7
WILVR	WKUP 引脚输入电平寄存器	8.8
DSRAMR	深度待机 RAM 保留寄存器	8.9
BUR01 至 16	备用寄存器 01 至 16	8.10
MOSC_CTL	主振荡驱动电平控制寄存器	8.11
WIOLC_CTL	IO 状态控制寄存器	8.12
SUBOSC_CTL	副振荡器 IO 控制寄存器	8.13
CEC_CTL	CEC IO 控制寄存器	8.14
DEBUG_SW_CTL	串行线调试控制寄存器	8.15

■ 低功耗模式所配备的各类型寄存器列表

Table 8-3 低功耗模式寄存器列表

寄存器名称	TYPE 1-M0+	TYPE 2-M0+	TYPE 3-M0+
STB_CTL	○	○	○
REG_CTL	○	-	○
RCK_CTL	○	○	○
PMD_CTL	○	○	○
WRFSR	-	○	○
WIFSR	-	○	○
WIER	-	○	○
WILVR	○	○	○
DSRAMR	-	○	○
BUR01 至 16	-	○	○
MOSC_CTL	-	○	○
WIOLC_CTL	-	-	○
SUBOSC_CTL	-	-	○
CEC_CTL	-	-	○

寄存器名称	TYPE 1-M0+	TYPE 2-M0+	TYPE 3-M0+
DEBUG_SW_CTL	-	-	○

注意事项：

- *有关系统时钟模式控制寄存器 (SCM_CTL) ，参见“时钟”一章。*
  深度待机模式时深度待机控制块的寄存器不关闭。

8.1 待机模式控制寄存器 (STB_CTL)

待机模式控制寄存器 (STB_CTL) 控制待机模式和深度待机模式。只有同时将 0x1ACC 写入 KEY 位，写入 SPL、DSTM 或 STM 位的值才有效。

寄存器配置

位	31														16	15							8
字段	KEY															保留							
属性	R/W															-							
初始值	0x0000															0x00							

位	7	6	5	4	3	2	1	0
字段	保留			SPL	保留	DSTM	STM	
属性	-			R/W	-	R/W	R/W	
初始值	000			0	0	0	00	

寄存器功能

[bit31:16] KEY：待机模式控制写入控制位

这些位释放 SPL 位、DSTM 位或 STM 位的写入控制。

- 只有将 0x1ACC 写入 KEY 位，写入 SPL 位、DSTM 位或 STM 位的值才有效。
- 如 KEY 位未写入 0x1ACC，则写入 SPL 位、DSTM 位或 STM 位的值无效。
- 读取模式下读取值总为 0x0000。

[bit15:5] 保留：保留位

读取值总为 "0"。操作时写入无效。

[bit4] SPL：待机模式引脚电平设置位

此位设置计时器模式、RTC 模式、停止模式、深度待机 RTC 模式或深度待机停止模式的引脚状态。

位	描述
0	保留计时器模式、RTC 模式及停止模式下各引脚的状态，并转换至深度待机 RTC 模式和深度待机停止模式下的 GPIO。[初始值]
1	设置计时器模式、RTC 模式、停止模式、深度待机 RTC 模式或深度待机停止模式下各引脚的状态为高阻抗状态。

[bit3] 保留：保留位
读取值总为 "0"。操作时写入无效。

[bit2] DSTM：深度待机模式选择位
此位选择转换至待机模式或深度待机模式。

[bit1:0] STM：待机模式选择位
这些位为 RTC 模式控制寄存器 (PMD_CTL) 的 DSTM 位和 RTCE 位的组合，选择转换至以下任一模式：计时器模式、RTC 模式、停止模式、深度待机 RTC 模式及深度待机停止模式。

DSTM	STM		PMD_CTL: RTCE	描述
	bit1	bit0		
0	0	0	0	计时器模式[初始值]
0	0	0	1	禁止设置。
0	0	1	0	禁止设置
0	0	1	1	禁止设置
0	1	0	0	STOP 模式
0	1	0	1	RTC 模式
0	1	1	0	禁止设置
0	1	1	1	禁止设置
1	0	0	0	禁止设置
1	0	0	1	禁止设置
1	0	1	0	禁止设置
1	0	1	1	禁止设置
1	1	0	0	深度待机 STOP 模式
1	1	0	1	深度待机 RTC 模式
1	1	1	0	禁止设置
1	1	1	1	禁止设置

注意事项：

- *只有同时将 "0x1ACC" 写入 KEY 位，写入待机模式控制寄存器 (STB_CTL) SPL 位、DSTM 位及 STM 位的值才有效。如 KEY 位未写入 "0x1ACC"，则写入 SPL 位、DSTM 位及 STM 位的值无效。*

8.2 副振荡电路电源控制寄存器 (REG_CTL)

副振荡电路电源控制寄存器 (REG_CTL) 控制副振荡电路的电源。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	保留					ISUBSEL		保留
属性			-			R/W		-
初始值			00000			10		0

寄存器功能

[bit7:3] 保留：保留位
读取值总为 "0"。操作时写入无效。

[bit2:1] ISUBSEL：副振荡电路电流设置位
这些位设置副振荡电路的电流。

bit1	bit0	描述
0	0	禁止设置
0	1	禁止设置
1	0	360nA [初始值]
1	1	禁止设置

[bit0] 保留：保留位
读取值总为 "0"。操作时写入无效。

注意事项：
- *软件复位或深度待机转换复位时，不初始化此寄存器。*
- *TYPE 2-M0+产品没有此寄存器。*

8.3 副时钟控制寄存器 (RCK_CTL)

副时钟控制寄存器 (RCK_CTL) 控制 RTC、HDMI-CEC/遥控接收的时钟。
停止向未使用的资源提供时钟可降低功耗。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	保留						CECCKE	RTCCKE
属性	-						R/W	R/W
初始值	000000						0	1

寄存器功能

[bit7:2] 保留：保留位
读取值总为 "0"。操作时写入无效。

[bit1] CECCKE：CEC 时钟控制位
此位控制 HDMI-CEC/遥控接收宏的副时钟。

位	描述
0	不向 HDMI-CEC/遥控接收宏提供副时钟。[初始值]
1	向 HDMI-CEC/遥控接收宏提供副时钟。

[bit0] RTCCKE：RTC 时钟控制位
此位控制 RTC 宏的副时钟。

位	描述
0	不向 RTC 宏提供副时钟。
1	向 RTC 宏提供副时钟。[初始值]

8.4 RTC 模式控制寄存器(PMD_CTL)

RTC 模式控制寄存器 (PMD_CTL) 控制 RTC 模式或停止模式以及深度待机 RTC 模式或深度待机停止模式。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	保留							RTCE
属性				-				R/W
初始值				0000000				0

寄存器功能

[bit7:1] 保留：保留位

读取值总为 "0"。操作时写入无效。

[bit0] RTCE：RTC 模式控制位

此位选择转换至 RTC 模式或停止模式，或转换至深度待机 RTC 模式或深度待机停止模式。

位	描述
0	停止模式和深度待机停止模式[初始值]
1	RTC 模式和深度待机 RTC 模式

DSTM 位为 "0" 时选择待机模式，DSTM 位为 "1" 时则选择深度待机模式。

注意事项：

- *软件复位和深度待机转换复位时，不初始化此寄存器。*
- *只有系统时钟模式状态寄存器 (SCM_STR) 的 SORDY 位为 "1"，将 "1" 写入 RTCE 位才有效。*
- *不管系统时钟模式控制寄存器 (SCM_CTL) 的 SOSCE 位以及系统时钟模式状态寄存器 (SCM_STR) 的 SORDY 位为何值，RTCE 位为 "1" 时使能副振荡。*

8.5 深度待机唤醒因素寄存器 1 (WRFSR)

深度待机唤醒因素寄存器 1 (WRFSR) 指示深度待机模式时的唤醒因素低压检测复位和 INITX 引脚输入复位。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	保留						WLVDH	WINITX
属性	-						R	R
初始值	000000						0	0

寄存器功能

[bit7:2] 保留：保留位

读取值总为 "0"。操作时写入无效。

[bit1] WLVHD：低压检测复位唤醒位

此位指示深度待机模式时的唤醒因素低压检测复位。

位	描述
0	没有低压检测复位唤醒[初始值]
1	低压检测复位唤醒

[bit0] WINITX：INITX 引脚输入复位唤醒位

此位指示深度待机模式时的 INITX 引脚输入复位唤醒。

位	描述
0	没有 INITX 引脚输入复位唤醒 [初始值]
1	INITX 引脚输入复位唤醒

注意事项：

- *上电复位及低压检测复位时，此寄存器初始化。其他复位因素时不初始化此寄存器。此外，读取可清除所有位。*
- *转换至深度待机模式前，确保未设置深度待机模式的唤醒因素。如设置了此类唤醒因素，将其清除。*
- *仅在深度待机模式下可设置此寄存器。*

8.6 深度待机唤醒因素寄存器 2 (WIFSR)

深度待机唤醒因素寄存器 2 (WIFSR) 指示深度待机模式时的唤醒因素 WKUPx 引脚输入、低压检测中断、RTC 中断及 HDMI-CEC/遥控接收中断。

寄存器配置

位	15	14	13	12	11	10	9	8
字段	WUI11	WUI10	WUI9	WUI8	WUI7	WUI6	WCEC1I	WCEC0I
属性	R	R	R	R	R	R	R	R
初始值	0	0	0	0	0	0	0	0

位	7	6	5	4	3	2	1	0
字段	WUI5	WUI4	WUI3	WUI2	WUI1	WUI0	WLVDI	WRTCI
属性	R	R	R	R	R	R	R	R
初始值	0	0	0	0	0	0	0	0

寄存器功能

[bit15:10] WUI11 至 WUI6：WKUPx 引脚输入唤醒位

这些位指示深度待机模式时的唤醒因素 WKUPx 引脚输入复位。

位	描述
0	没有 WKUPx 引脚输入唤醒 [初始值]
1	WKUPx 引脚输入唤醒

[bit9] WCEC1I：CEC ch.1 中断唤醒位

此位指示深度待机模式时的 HDMI-CEC/遥控接收 ch.1 中断唤醒。

位	描述
0	没有 HDMI-CEC/遥控接收 ch.1 中断未唤醒[初始值]
1	HDMI-CEC/遥控接收 ch.1 中断唤醒

[bit8] WCEC0I：CEC ch.0 中断唤醒位

此位指示深度待机模式时的 HDMI-CEC/遥控接收 ch.0 中断唤醒。

位	描述
0	没有 HDMI-CEC/遥控接收 ch.0 中断唤醒[初始值]
1	HDMI-CEC/遥控接收 ch.0 中断唤醒

[bit7:2] WUI5 至 WUI0：WKUPx 引脚输入唤醒位

这些位指示深度待机模式时的唤醒因素 WKUPx 引脚输入。

位	描述
0	没有 WKUPx 引脚输入唤醒 [初始值]
1	WKUPx 引脚输入唤醒

[bit1] WLVDI：LVD 中断唤醒位

此位指示通过 LVD 中断从深度待机模式唤醒。

位	描述
0	没有 LVD 中断唤醒[初始值]
1	LVD 中断唤醒

[bit0] WRTCI：RTC 中断唤醒位

此位指示深度待机模式时的唤醒因素 RTC 中断。

位 t	描述
0	没有 RTC 中断唤醒[初始值]
1	RTC 中断唤醒

注意事项：

- *上电复位及低压检测复位时，此寄存器初始化。其他复位因素时不初始化此寄存器。此外，读取可清除所有位。*
- *转换至深度待机模式前，确保未设置深度待机模式的唤醒因素。如设置了此类唤醒因素，将其清除。*
- *仅在深度待机模式下可设置此寄存器。*

8.7 深度待机唤醒使能寄存器 (WIER)

深度待机唤醒使能寄存器 (WIER) 使能通过 WKUPx 引脚输入、低压检测中断、RTC 中断及 HDMI-CEC/遥控接收中断从深度待机模式唤醒。

寄存器配置

位	15	14	13	12	11	10	9	8
字段	WUI11E	WUI10E	WUI9E	WUI8E	WUI7E	WUI6E	WCEC1E	WCEC0E
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

位	7	6	5	4	3	2	1	0
字段	WUI5E	WUI4E	WUI3E	WUI2E	WUI1E	保留	WLVDE	WRTCE
属性	R/W	R/W	R/W	R/W	R/W	-	R/W	R/W
初始值	0	0	0	0	0	0	0	0

寄存器功能

[bit15:10] WUI11E 至 WUI6E：WKUPx 引脚输入唤醒使能位

禁用或使能 WKUPx 引脚输入唤醒深度待机模式。

位	描述
0	禁用 WKUPx 引脚输入唤醒 [初始值]
1	使能 WKUPx 引脚输入唤醒

[bit9] WCEC1E：HDMI-CEC/遥控接收 ch.1 中断唤醒使能位

禁用或使能通过 HDMI-CEC/遥控接收 ch.1 中断从深度待机模式唤醒。

位	描述
0	禁用 HDMI-CEC/遥控接收 ch.1 中断唤醒[初始值]
1	使能 HDMI-CEC/遥控接收 ch.1 中断唤醒

[bit8] WCEC0E：HDMI-CEC/ 遥控接收 ch.0 中断唤醒使能位

禁用或使能通过 HDMI-CEC/遥控接收 ch.0 中断从深度待机模式唤醒。

位	描述
0	禁用 HDMI-CEC/遥控接收 ch.0 中断唤醒[初始值]
1	使能 HDMI-CEC/遥控接收 ch.0 中断唤醒

[bit7:3] WUI5E 至 WUI1E：WKUPx 引脚输入唤醒使能位

禁用或使能通过 WKUPx 引脚输入从深度待机模式唤醒。

位	描述
0	禁用 WKUPx 引脚输入唤醒 [初始值]
1	使能 WKUPx 引脚输入唤醒

[bit2] 保留：保留位

读取值总为 "0"。操作时写入无效。

[bit1] WLVDE：LVD 中断唤醒使能位

禁用或使能通过 LVD 中断从深度待机模式唤醒。

位	描述
0	禁用 LVD 中断唤醒[初始值]
1	使能 LVD 中断唤醒

[bit0] WRTCE：RTC 中断唤醒使能位

禁用或使能通过 RTC 中断从深度待机模式唤醒。

位	描述
0	禁用 RTC 中断唤醒[初始值]
1	使能 RTC 中断唤醒

注意事项：

- *始终使能 WKUP0 引脚输入从深度待机模式唤醒。*
- *深度待机转换复位时不初始化此寄存器。*

8.8 WKUP 引脚输入电平寄存器 (WILVR)

WKUP 引脚输入电平寄存器 (WILVR) 选择深度待机模式时的 WKUP1 至 WKUP5 引脚输入有效电平。

寄存器配置

位	15	14	13	12	11	10	9	8
字段	保留					WUI11LV	WUI10LV	WUI9LV
属性			-			R/W	R/W	R/W
初始值			00000			0	0	0

位	7	6	5	4	3	2	1	0
字段	WUI8LV	WUI7LV	WUI6LV	WUI5LV	WUI4LV	WUI3LV	WUI2LV	WUI1LV
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

寄存器功能

[bit15:11] 保留：保留位

读取值总为 "0"。操作时写入无效。

[bit10:0] WUI11LV 至 WUI1LV：WKUPx 引脚输入电平选择位

选择 WKUPx 引脚输入的有效电平。

位	描述
0	WKUPx 引脚输入为低电平时请求唤醒[初始值]
1	WKUPx 引脚输入为高电平时请求唤醒

注意事项：

- *低电平时 WKUP0 引脚输入始终请求唤醒*
  例如，转换为深度待机模式时如果 WKUP1 输入为低电平 (WUI1LV = 0) ，则立即唤醒。
- *深度待机转换复位不初始化此寄存器。*

8.9 深度待机 RAM 保留寄存器 (DSRAMR)

深度待机 RAM 保留寄存器 (DSRAMR) 控制深度待机模式时保留片上 SRAM 的内容。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	保留						SRAMR	
属性	-						R/W	
初始值	000000						00	

寄存器功能

[bit7:2] 保留：保留位

读取值总为 "0"。操作时写入无效。

[bit1:0] SRAMR：片上 SRAM 保留控制位

这些位控制深度待机模式时保留片上 SRAM 的内容。

bit1	bit0	描述
0	0	不保留深度待机模式时片上 SRAM 的内容。[初始值]
0	1	禁止设置
1	0	禁止设置
1	1	保留深度待机模式时片上 SRAM 的内容

注意事项：

- *上电复位及低压检测复位时，此寄存器初始化。其他复位因素时不初始化此寄存器。*

8.10 备用寄存器 01 至 16 (BUR01 至 16)

备用寄存器 01 至 16（BUR01 至 16）为通用寄存器，用于深度待机模式时保留值。

寄存器配置

位	31	24	23	16	15	8	7	0
字段	BUR04		BUR03		BUR02		BUR01	
属性	R/W		R/W		R/W		R/W	
初始值	0x00		0x00		0x00		0x00	

位	31	24	23	16	15	8	7	0
字段	BUR08		BUR07		BUR06		BUR05	
属性	R/W		R/W		R/W		R/W	
初始值	0x00		0x00		0x00		0x00	

位	31	24	23	16	15	8	7	0
字段	BUR12		BUR11		BUR10		BUR09	
属性	R/W		R/W		R/W		R/W	
初始值	0x00		0x00		0x00		0x00	

位	31	24	23	16	15	8	7	0
字段	BUR16		BUR15		BUR14		BUR13	
属性	R/W		R/W		R/W		R/W	
初始值	0x00		0x00		0x00		0x00	

注意事项：

- *上电复位及低压检测复位时，此寄存器初始化。其他复位因素时不初始化此寄存器。*

8.11 主晶振型选择控制寄存器 (MOSC_CTL)

主晶振型选择控制寄存器 (MOSC_CTL) 用于选择外部晶振的类型。配备适当的晶振类型可降低功耗。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	保留					IMAINSEL		保留
属性			-			R/W		-
初始值			00000			10		0

寄存器功能

[bit7:3] 保留：保留位

读取值总为 "0"。操作时写入无效。

[bit2:1] IMAINSEL：主振荡电路电流设置位

这些位设置主振荡电路的电流。

TYPE 2-M0+：

bit1	bit0	描述
0	0	仅支持 4MHz 晶振。最低功耗。
0	1	支持 4MHz/8MHz 晶振，中等功耗。
1	0	支持 4MHz/8MHz/16MHz 晶振，较高功耗[初始设置]
1	1	仅支持 48MHz 晶振，最高功耗。

TYPE 3-M0+：

bit1	bit0	描述
0	0	禁止设置。
0	1	最高支持 8MHz 晶振，中等功耗。
1	0	支持 16MHz 晶振，较高功耗[初始设置]
1	1	最高支持 48MHz 晶振，最高功耗。

[bit0] 保留：保留位

读取值总为 "0"。操作时写入无效。

注意事项：

- *软件复位或深度待机转换复位时，不初始化此寄存器。*

8.12 IO 状态保持控制寄存器 (WIOLC_CTL)

IO 状态保持控制寄存器 (WIOLC_CTL) 设置深度待机模式下 IO 状态保持功能的特性，并释放 IO 状态保持。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	保留							保留
属性	-							-
初始值	0000000							0

位	23	22	21	20	19	18	17	16
字段	保留							LHX_ST
属性	-							R
初始值	0000000							1

位	15	14	13	12	11	10	9	8
字段	保留							CONTX
属性	-							R/W
初始值	0000000							0

位	7	6	5	4	3	2	1	0
字段	保留							LH_CL
属性	-							R/W
初始值	0000000							0

寄存器功能

[bit31:25] 保留：保留位
这些位读为 "0x00"。
写入这些位时，将其设置为 "0x00"。

[bit24] 保留：保留位
这些位读为 "0x0"。
写入此位时，将其设置为 "0x0"。

[bit23:17] 保留：保留位
这些位读为 "0x00"。
写入这些位时，将其设置为 "0x00"。

[bit16] LHX_ST：IO 状态位
指示 IO 状态为保持或释放。

Bit16		描述
读取	0	IO 状态保持
	1	IO 状态释放（直通）

[bit15:9] 保留：保留位

这些位读为 "0x00"。

写入这些位时，将其设置为 "0x00"。

[bit8] CONTX：IO 状态保持功能使能位

使能 IO 状态保持功能。

Bit8		描述
读取		读取寄存器值。
写入	0	使能 IO 保持功能
	1	禁用 IO 保持功能

[bit7:1] 保留：保留位

这些位读为 "0x00"。

写入这些位时，将其设置为 "0x00"。

[bit0] LH_CL：IO 状态保持释放位

释放 IO 状态保持。 写入此位时，必须运行高速 CR 时钟。

Bit8		描述
读取		总是读为 ‘0’
写入	0	IO 状态未受影响。
	1	释放 IO 状态（运行模式）。

8.13 副振荡器 IO 控制寄存器 (SUBOSC_CTL)

SUBOSC_CTL 寄存器将引脚设置为副时钟（振荡）引脚。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	保留						SUBXC	
属性	-						R/W	
初始值	-						01	

寄存器功能

[bit7:2] 保留：保留位

这些位读为 "0"。

写入这些位时，将其设置为 "0"。

[bit1:0] SUBXC：副时钟（振荡）引脚设置寄存器

此位将引脚设置为副时钟（振荡）引脚。

bit1:0		描述
读取		读取寄存器值。
写入	00	X0A 和 X1A 引脚不作为副时钟（振荡）引脚，而是作为数字输入/输出引脚。
	01	X0A 和 X1A 引脚作为副时钟（振荡）引脚。[初始值]（I/O 单元将处于输入方向、输入切断和上拉断开状态。）
	10	禁止设置。
	11	深度待机模式： 禁止设置。 其他模式： X0A 引脚作为外部时钟输入引脚。 X1A 引脚用于数字输入/输出。

注意事项：

- *仅 SUBXC 位写"01" 不能起振副时钟。*
  要启动振荡，应在 SUBXC 位写入 "01" 后，通过如 "时钟" 一章所述系统时钟模式控制寄存器 (SCM_CTL) 的 SOSCE 位使能振荡。
- *有关外部时钟的使用，参见各产品《数据手册》"处理注意事项"中"外部时钟的使用"。*
- *深度待机转换复位时不初始化此寄存器。*

8.14 CEC 输入/输出控制寄存器 (CEC_CTL)

CEC 输入/输出控制寄存器 (CEC_CTL) 用于选择 HDMI-CEC/遥控接收的 I/O 端口。此寄存器仅用于 TYPE 3-M0+产品。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	保留				WS_CECR1B		WS_CECR0B	
属性					R/W		R/W	
初始值	0000				00		00	

寄存器功能

[bit7:4] 保留：保留位

读取值总为 "0"。操作时写入无效。

[bit3:2] WS_CECR1B：CEC1 输入/输出选择位

此位用于选择 HDMI-CEC/遥控接收 ch.1 的 IO 端口。

位		描述
读取		读取寄存器值
写入	00	不从 HDMI-CEC/遥控接收 ch.1 输入/输出。[初始值]
	01	使用 CEC1_0，从 HDMI-EC/遥控接收 ch.1 输入/输出
	10	禁止设置
	11	禁止设置

[bit1:0] WS_CECR0B：CEC0 输入/输出选择位

此位用于选择 HDMI-CEC/遥控接收 ch.0 的 IO 端口。

位		描述
读取		读取寄存器值
写入	00	不从 HDMI-CEC/遥控接收 ch.0 输入/输出。[初始值]
	01	使用 CEC0_0，从 HDMI-EC/遥控接收 ch.0 输入/输出
	10	禁止设置
	11	禁止设置

注意事项：

- *在 TYPE 3-M0+ 产品中，此寄存器的设置有效，EPFR18 寄存器的 CECR1B/CECR0B 无效。写入 EPFR18 寄存器的 CECR1B/CECR0B 时，此位总是写 0 。优先设置此寄存器。*
- *如 WS_CECR1B/WS_CECR0B 设置为 “01”，无论 PCR 的设置如何，CEC 的上拉电阻关闭。*
- *深度待机转换复位时不初始化此寄存器。*

8.15 串行线调试控制寄存器 (DEBUG_SW_CTL)

串行线调试控制寄存器 (DEBUG_SW_CTL) 用于设置串行线调试的终端。

寄存器配置

位	7	6	5	4	3	2	1	0
字段	保留							DBG_EN
属性				R/W				R/W
初始值				0000000				1

寄存器功能

[bit7:1] 保留：保留位

读取值总为 "0"。操作时写入无效。

[bit0] DBG_EN：SWD 使能位

此位使能串行线调试。

Bit8		描述
读取		读取寄存器值。
写入	0	使用 GPIO 端口
	1	使用串行线调试端口

注意事项：

- *使能 SWD 时，除使能 DBG_EN 外，还必须将 EPFR00 寄存器的 SWDEN 位设置为 1。*
- *SWD 端口（串行线调试端口）作为 GPIO 端口时，将此位设置为 0.*
- *深度待机转换复位不初始化此寄存器。*

9. 使用注意事项

使用低功耗模式时需注意以下几点：

如果是复作为模拟输入和 WKUP 的引脚，对应模拟输入设置寄存器 (ADE) 的 ADE 位设置为 "1" 时，即便允许通过 WKUPx 引脚输入唤醒，WKUPx 引脚输入也会被屏蔽。要通过 WKUPx 引脚输入唤醒，应在切换至深度待机模式前，将对应模拟输入设置寄存器 (ADE) 的 ADE 位设置为 "0"。

第 6-2 章：Vbat 域

本章说明 **VBAT** 电源域的功能和操作。

1. **VBAT 域概述**
2. **VBAT 域配置**
3. **芯片电源控制**
4. **睡眠控制**
5. **32 kHz 时钟设置步骤**
6. **VBAT I/O 端口设置步骤**
7. **寄存器**
8. **使用注意事项**

代码：9BFVBATPD-FM0-C03.0

1. VBAT 域概述

采用 VBAT 电源引脚为 RTC（日历电路）和 32 kHz 振荡器提供独立的电源，可降低 RTC 运行过程中的功耗。适用于 TYPE 2-M0+产品。

电源域的配置

本产品族包括下列三个电源域：

■ CPU 域

本域由下列电路构成：

- CPU
- 片上闪存
- 片上 SRAM
- 外设功能

本域通过 VCC 电源引脚供电。在深度待机 RTC 模式和深度待机停止模式下该电源断开。

■ 常开域

本域由下列电路构成：

- 片上调节器
- 电源管理电路
- 端口电路
- 主振荡电路和 I/O 端口
- RTC（控制功能、计时器）

本域通过 VCC 电源引脚供电。

VCC 电源引脚通过系统电源供电（板载调节器）。

■ VBAT 域

本域由下列电路构成：

- RTC（日历功能）
- 32 kHz 振荡电路
- 上电电路
- 备用寄存器
- 端口电路

此域总是由 VBAT 电源引脚供电。

VBAT 电源引脚通过备用电源（例如电池）和系统电源供电。

片上电源门控

在深度待机 RTC 模式和深度待机停止模式下，本族产品通过芯片内置电源开关功能元件切断 CPU 的电源。

即使是在深度待机 RTC 模式和深度待机停止模式下常开域也可以使电源保持接通。

有关深度待机 RTC 模式和深度待机停止模式的详细信息，参见“低功耗模式”一章。

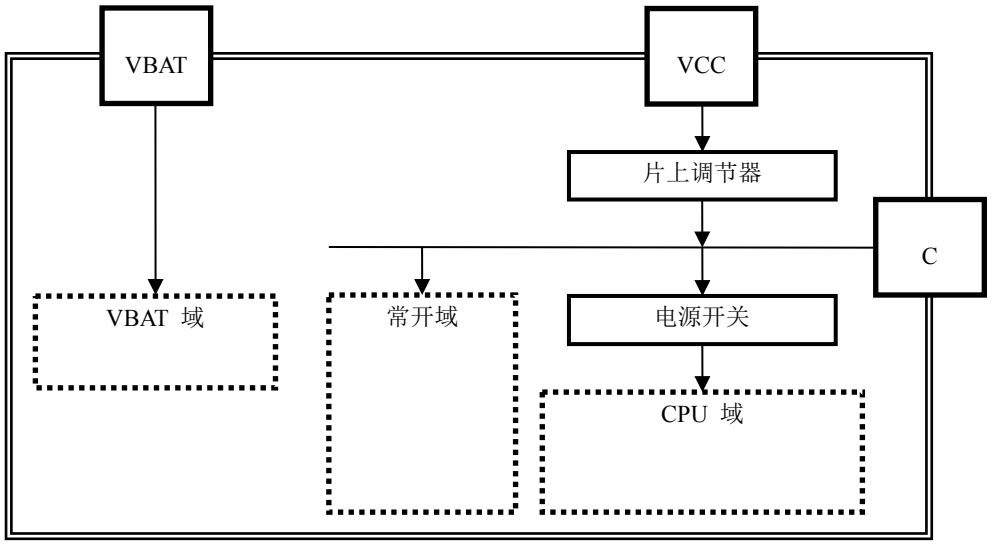
片外电源门控

当本族产品的系统电源供电被切断时，CPU 域和常开域的电源会被切断。

在这种情况下，如果有备用电源向 VBAT 域供电，则 VBAT 域不会断电。

板载调节器（通过 RTC 内的报警时间设置或通过外部引脚发出的信号输入控制系统电源）可以开启或关闭。

Figure 1-1 FM0+ 家族的电源配置



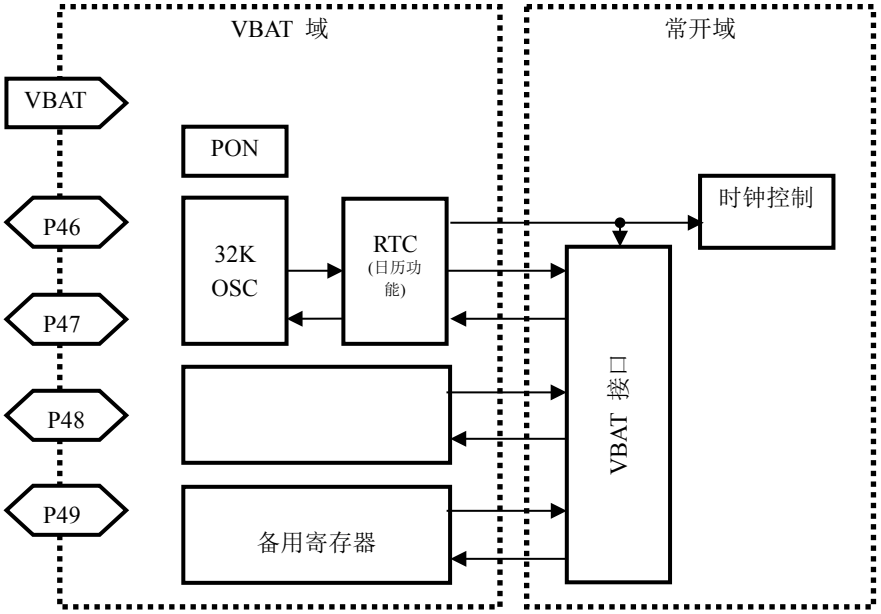
2. VBAT 域配置

本节说明 VBAT 域的内部配置。

VBAT 域的内部配置

Figure 2-1 所示为 VBAT 域的内部配置以及 VBAT 域和常开域之间的连接。

Figure 2-1 VBAT 域的内部配置及 VBAT 域和常开域之间的连接



■ RTC（日历功能）

这是一个带频率补偿功能的日历电路。不包括定时功能。

■ 32 kHz 振荡电路 (32K OSC)

这个振荡电路可连接至 (32768 Hz) 的时钟晶振器。

■ 上电电路(PON)

此电路用于检测 VBAT 域的上电情况并生成电路初始化信号。

■ 备用寄存器

本寄存器为 32 字节寄存器，用于在向 VBAT 电源引脚供电时保持数据。

■ VBAT I/O 端口（P46 至 49，端口控制）

这些 I/O 端口通过 VBAT 电源引脚供电而被驱动。

VBAT I/O 端口的控制电路与用于 I/O 端口（除 P46 至 P49 外）之外的其他控制电路是相互独立的。

2.1 与常开域的接口

本节说明 VBAT 域和常开域之间的接口方法。

接口概述

VBAT 域由 32 kHz 振荡电路或从 PCLK 分频的时钟驱动。

因此，如果某条内部总线直接与 VBAT 域内的某个寄存器连接，则在访问这个寄存器时会有一条总线主控器（如 CPU）会被迫等待。

FM0+家族产品可通过以下两种机制来防止被迫等待总线的访问：

- 在常开域中内置一个缓冲器，从内部总线访问被指向该缓冲器。
- 数据在常开域的缓冲器和 VBAT 域的寄存器之间传输。

在《FM0+家族产品手册》中，有关常开域的缓冲器和 VBAT 域的寄存器之间数据传输操作的调用描述如下：

- 回调：数据从 VBAT 域的寄存器传输到常开域的缓冲器。
- 保存：数据从常开域的缓冲器传输到 VBAT 域的寄存器。

VCC 电源断开时常开域缓冲器中写入的数据会被擦除，因此应在 VCC 电源开启时将数据保存到 VBAT 域的寄存器内。

VCC 电源接通后或常开域中发生复位时，缓冲器内的值会立即被常开域复位信号初始化。

因此，在从缓冲器读取数据之前，应在 VBAT 电源（备用电源）接通时执行回调操作将寄存器内保留的数据恢复到缓冲器中。

缓冲器内的 RTC 日历数据不会自动更新。

在从缓冲器中读取时间数据之前，应执行回调操作将 VBAT 域寄存器内保存的时间数据传输到缓冲器中。

接口电路类型

如 Table 2-1 所示，接口电路有四种类型。

Table2-1 接口电路类型

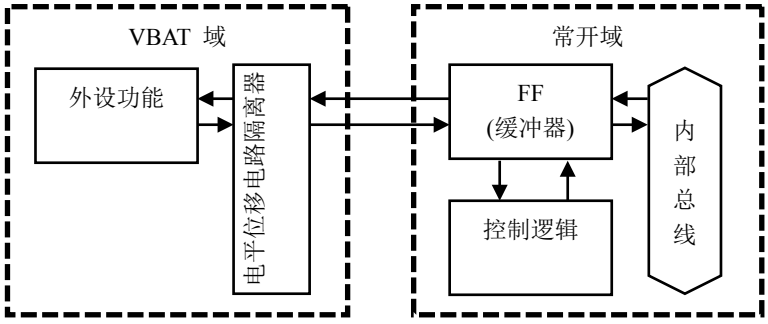
电路类型	常开域	VBAT 域	传输时钟	图号
TYPE 1	有 FF	FF 无 FF	-	Figure 2-2
TYPE 2	有 FF	有 FF	32 kHz	Figure 2-3
TYPE 3	有 FF	有 FF	PCLK1（分频）	Figure 2-4
TYPE 4	FF 无 FF	有 FF	-	Figure 2-5

当常开域断电时，从常开域发送到 VBAT 域的信号被电平位移器和隔离器截取至 VSS。

有了这个功能就可以在常开域断电时执行以下操作：继续执行 RTC 的日历功能和报警功能相关操作、保持 VBAT I/O 端口的引脚状态、将数据保存在备用寄存器中。

■ 接口电路-TYPE 1

Figure 2-2 TYPE 1 接口电路配置



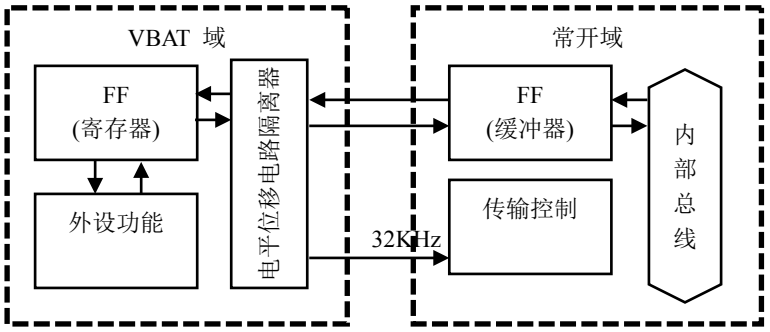
如果寄存器不需要在 VCC 电源断开时保留数据，就可以使用这种类型的接口电路。

Table 2-2 TYPE 1 接口电路寄存器的特性

	寄存器性态
寄存器初始化	寄存器通过常开域复位信号初始化。
总线读取	可直接读取控制电路（常开域）和外设电路（VBAT 域）的状态。
总线写入	寄存器直接影响控制电路（常开域）和外设电路（VBAT 域）的操作。

■ 接口电路 TYPE 2

Figure 2-3 接口电路 TYPE 2 的配置



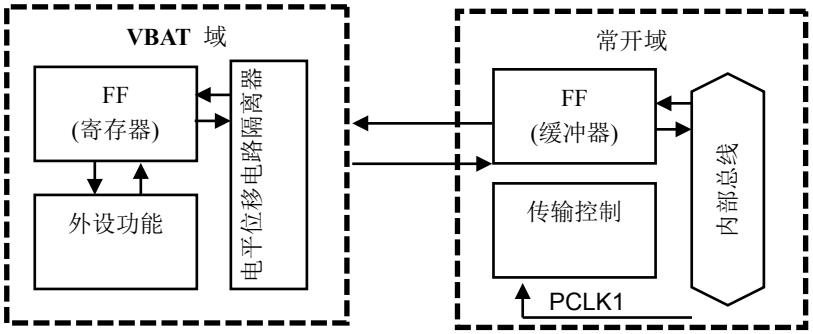
如果寄存器需要在 VCC 电源断开时保留数据，应使用这种类型的接口电路。

Table 2-3TYPE 2 接口电路寄存器的特性

	寄存器/缓冲器性态
寄存器初始化	寄存器通过 VBAT 域的上电信号初始化。
缓冲器初始化	寄存器通过常开域复位信号初始化。 有关复位因素，参见“RTC 计数模块”一章。
总线读取	将缓冲器中的数据读取至总线。
总线写入	将数据写入缓冲器。
回调操作	数据从寄存器传输到缓冲器。
保存操作	数据从缓冲器传输到寄存器。

■ 接口电路-TYPE 3

Figure 2-4 TYPE 3 接口电路配置



如果寄存器需要在 VCC 电源断开时保留数据，应使用这种类型的接口电路。

Table 2-4 TYPE 3 接口电路寄存器的特性

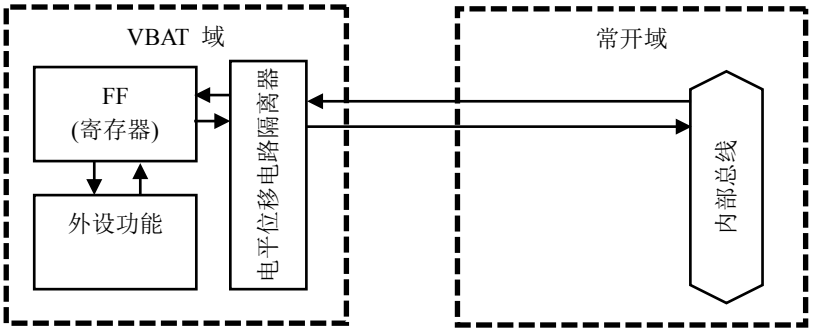
	寄存器/缓冲器特性
寄存器初始化	寄存器通过 VBAT 域的上电信号初始化。

缓冲器初始化	寄存器通过常开域复位信号初始化。 寄存器通过 RTC 复位初始化。
总线读取	将缓冲器中的数据读取至总线。
总线写入	将数据写入缓冲器。
回调操作	数据从寄存器传输到缓冲器。
保存操作	数据从缓冲器传输到寄存器。

TYPE 2 接口电路和 TYPE 3 接口电路的区别在于回调操作和保持操作的时钟。

■ 接口电路 TYPE 4

Figure 2-5 TYPE 4 接口电路配置



如果寄存器需要在 VCC 电源断开时保留数据，应使用这种类型的接口电路。

Table 2-5 TYPE 4 接口电路寄存器特性

	寄存器/缓冲器特性
寄存器初始化	寄存器通过 VBAT 域的上电信号初始化。
总线读取	将缓冲器中的数据读取至总线。
总线写入	直接影响 VBAT 域的操作。

TYPE 4 接口电路不需要回调操作或保存操作。

连接接口电路的电路

VBAT 域内的主要电路包括 RTC（日历功能）、VBAT 端口和缓冲器寄存器。

VBAT 域对各个电路的缓冲器和寄存器执行保存操作或回调操作。

有关 WTCR20 寄存器的功能的详细说明，参见“计时器部分”中“RTC 计数模块”一章的“7.5 控制寄存器 (WTCR20)”。

- CREAD/CWRITE

对 RTC 电路中的寄存器（Table 2-6 中所示的）执行批量保存/回调操作。

Table 2-6 通过 CWRITE/CREAD 进行传输的寄存器列表

编号	寄存器名称	参考章节	编号	寄存器名称	参考章节
1	WTSR	[RTCCAL]	2	WTMIR	[RTCCAL]
3	WTHR	[RTCCAL]	4	WTDR	[RTCCAL]
5	WTDW	[RTCCAL]	6	WTMOR	[RTCCAL]
7	WTYR	[RTCCAL]	8	ALMIR	[RTCCAL]
9	ALHR	[RTCCAL]	10	ALDR	[RTCCAL]
11	ALMOR	[RTCCAL]	12	ALYR	[RTCCAL]
13	保留	-	14	WTCR11	[RTCCAL]
15	WTCR10（只限于 bit0）	[RTCCAL]			

有关各个寄存器的功能，参见 [RTCCAL]（表示“计时器部分”的“RTC 计数模块”一章）。

1 至 14 号寄存器的接口电路类型为 TYPE 2。

第 15 号 WTCR10 寄存器采用不同位数的接口电路。Bit:0 ST 为 TYPE 2。Bit:2 RUN 为 TYPE 4。除 bit0,2 外，本寄存器的数位为正常寄存器位，不受 VBAT 域的影响。

当 WTCR20 寄存器的 bit1 写入 "1"时，启动保存操作。这项保存操作称为 CWRITE 操作。

当 WTCR20 寄存器的 bit0 写入 "1"时，启动回调操作。此回调操作称为 CREAD 操作。

副时钟用作传输时钟。

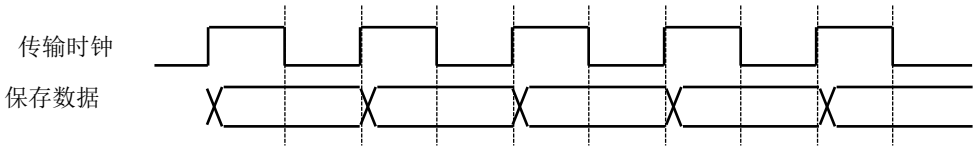
RTC 在一个传输时钟内传输 1 字节的数据。

一次 CREAD/CWRITE 操作中，RTC 按照 Table 2-6 中第 1 到 15 的顺序传输寄存器的数据。

因为寄存器总共 15 个字节，当 15 个传输时钟过去后数据传输结束。

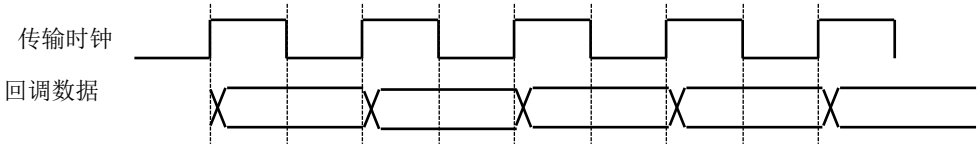
保存操作和回调操作过程中有一些需特别注意的事项。具体事项参见“计时器部分”的“RTC 计数模块”一章中“3.RTC 计数模块操作说明及设置步骤示例”。

- CWRITE 操作波形



保存数据在传输时钟的上升沿从缓冲器中输出，然后在传输时钟的下降沿写入寄存器。
有三个传输时钟需要在数据传输开始前进行预处理，有两个传输时钟还需要在传输结束后进行后处理。

- CREAD 操作波形



回调的数据在传输时钟的上升沿从寄存器输出，然后在所有数据传输结束时写入缓冲器。
有三个传输时钟需要在数据传输开始前进行预处理，有两个传输时钟还需要在传输结束后进行后处理。

■ PWRITE/PREAD

对 Table 2-7 所示 VBAT 端口电路中的寄存器进行批量保存/回调操作。

Table 2-7 通过 PWRITE/PREAD 进行传输的寄存器列表

编号	寄存器名称	参考章节	编号	寄存器名称	参考章节
1	WTCAL0	[RTCCLK]	2	WTCAL1	[RTCCLK]
3	WTCALPRD	[RTCCLK]	4	WTCALEN	[RTCCLK]
5	WTCOSEL	[RTCCLK]	6	CCS	2.332 kHz 振荡电路
7	保留	-	8	WTOSCCNT	2.332 kHz 振荡电路
9	VBPFR	2.6VBAT I/O 端口	14	VBPCR	2.6VBAT I/O 端口
11	VBDDR	2.6VBAT I/O 端口	12	VBPZR	2.6VBAT I/O 端口
13	VBDOR	2.6VBAT I/O 端口			

有关各个寄存器的功能，参见 [RTCCLK]（指“计时器部分”的“RTC 时钟控制模块”一章，以及本章的“2.6 VBAT I/O 端口”和“2.3 32 kHz 振荡电路”。

VBAT 端口电路中的第 1 至 13 号寄存器所用的接口电路为 TYPE 3。

当 WTCR20 寄存器的 bit5 写入 "1"时，执行保存操作。这项保存操作称为 PWRITE 操作。

当 WTCR20 寄存器的 bit4 写入 "1"时，执行回调操作。这项回调操作称为 PREAD 操作。

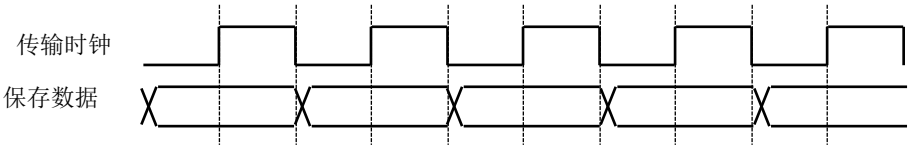
传输时钟是通过 PCLK1 除以 VB_CLKDIV 寄存器值生成。

RTC 在一个传输时钟内传输 1 字节的数据。

在执行 PREAD/PWRITE 操作时，RTC 寄存器按照 Table 2-7 中第 1 到 13 的顺序对寄存器的数据进行传输。

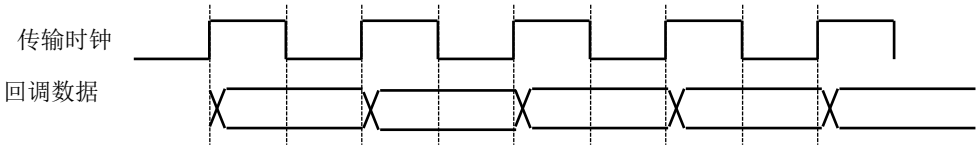
VBAT 端口电路的寄存器容量为 13 字节，因此数据传输会在 13 个传输时钟过去后结束。

- PWRITE 操作波形



保存的数据在传输时钟的下降沿从缓冲器中输出，然后在传输时钟的上升沿写入寄存器。
有一个传输时钟需要在数据传输开始前进行预处理。

- PREAD 操作波形



回调的数据在传输时钟的上升沿从寄存器中输出，然后在传输时钟的下降沿写入缓冲器。
有一个传输时钟需要在数据传输开始前进行预处理，有一个传输时钟还需要在数据传输结束后进行后处理。

■ BWRITE/BREAD

备用寄存器指 BREG00 至 BREG1F 的 32 字节寄存器区。

有关备用寄存器的功能，参见“2.5 备用寄存器”。

备用寄存器的接口电路类型为 TYPE 3。

当 WTCR20 寄存器的 bit3 写入 "1"时，启动保存操作。这项保存操作称为 BWRITE 操作。

当 WTCR20 寄存器的 bit2 写入 "1"时，启动回调操作。这项回调操作称为 BREAD 操作。

传输时钟是通过 PCLK1 除以 VB_CLKDIV 寄存器的值来生成的。

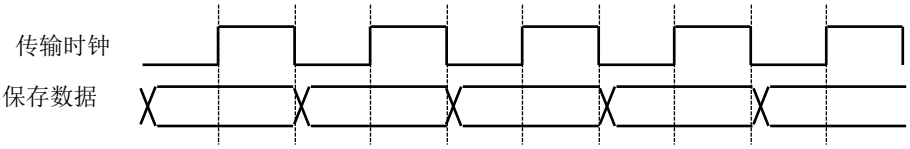
RTC 在一个传输时钟内传输 1 字节的数据。

在执行 BREAD/BWRITE 操作时，RTC 寄存器按照 BREG00 至 BREG1F 的顺序对寄存器的数据进行传输。

RTC 从 BREG00 开始传输。每当一个传输时钟结束时传输目的地或传输源转换至下一个寄存器。

FM0+家族的备用寄存器容量为 32 字节，因此数据传输会在 32 个传输时钟过去后结束。

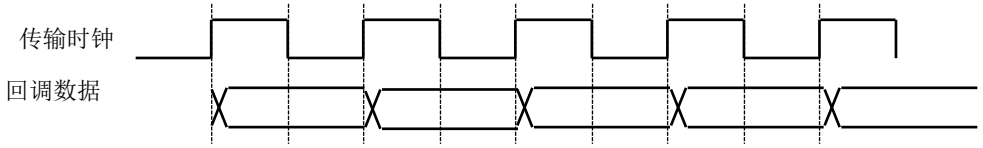
BWRITE 波形



保存的数据在传输时钟的下降沿从缓冲器中输出，然后在传输时钟的上升沿写入寄存器。

有一个传输时钟需要在数据传输开始前进行预处理。

- BREAD 操作波形



回调的数据在传输时钟的上升沿从寄存器中输出，然后在传输时钟的下降沿写入缓冲器。

有一个传输时钟需要在数据传输开始前进行预处理，有一个传输时钟还需要在数据传输结束后进行后处理。

■ 允许传输组合

尽管在回调或保存操作之前应检查 WTCR0 寄存器的 TRANS 位是否设置为 "0" ，下表中标记有 "o"的传输项可同时执行。

	CREAD	CWRITE	PREAD	PWRITE	BREAD	BWRITE
CREAD		×	×	×	o	o
CWRITE	×		×	×	o	o
PREAD	×	×		×	o	o
PWRITE	×	×	×		o	o
BREAD	o	o	o	o		×
BWRITE	o	o	o	o	×	

"o"表示这些传输操作可同时执行。

"×"表示这些传输操作不可同时执行。

■ 注释：

在《FM0+家族外设手册》中，对 TYPE 2 或 TYPE 3 接口电路的寄存器进行读取访问和写入访问的操作说明描述如下：

读取访问：执行回调操作，然后读取缓冲器中的数据。

写入访问：先执行回调操作对整个缓冲器的数据进行更新。然后，缓冲器中与写入访问相对应的数据会被此类数据替代，最后执行保存操作。

■ 使用注意事项

- 在进行保存操作或回调操作时不可访问缓冲器。如果在保存进程中访问缓冲器，无法确定数据是否正确保存；同样地，在回调操作时访问缓冲器，也无法确定读取的数据是否正确。
- 设置 VB_CLKDIV 寄存器值时，应使得分频 PCLK1 所生成的 PREAD、PWRITE、BREAD 及 BWRITE 的传输时钟频率为 1 MHz 或更低。

2.2 RTC

FM0+家族的 RTC 为具有 32 kHz 频率补偿功能的日历电路。

RTC 功能概述

RTC 具有以下功能：

- 时钟功能
- 报警功能
- 计时器功能（存在于常开域中）
- 频率补偿功能

RTC 配置

有关 RTC 功能的详细信息，参见“计时器部分”的“RTC 计数模块”一章。

2.3 32 kHz 振荡电路

32 kHz 振荡电路为时钟晶振器的专用振荡电路，用于生成副时钟。

32 kHz 振荡电路的功能概述

32 kHz 振荡电路具有以下功能：

- 振荡电流开关功能（存在于常开域中）
- 振荡增压功能（存在于常开域中）
- 时钟发生器协同操作功能

■ 振荡电流开关功能

32 kHz 振荡电路的放大器电路由恒流源供电。

恒流源的电流值可通过 CCS 寄存器的值进行控制。

■ 振荡增压功能

用于时钟晶振器时，振荡频率需更长的时间才能稳定下来。

在振荡启动后的一定时间内增加放大器电路的电流可以缩短振荡稳定所需的时间。

在振荡开始到 BOOST 寄存器中设置的振荡增压时间结束这段时间中，恒流源所供电流为 CCB 寄存器中所设电流值。以上时间过去后，恒流源所供电流切换至 CCS 寄存器中所设电流值。

当 CCB 寄存器中设定的电流值大于 CCS 寄存器中的电流设定值时，振荡增压功能能有效工作。

如果不使用振荡增压功能，则将 CCB 寄存器的电流值设定为与 CCS 寄存器中所设值相同。

■ 时钟发生器协同操作功能

WTOSCCNT 寄存器中的 SOSCNTL 位用于使能或禁用 32 kHz 振荡电路与 CPU 域的时钟发生器之间的协同操作。

使能协同操作功能后，当 CPU 转换至停止模式或深度待机停止模式时，本电路的 32 kHz 振荡停止。

当协同操作功能被禁用时，不管 CPU 处于哪种操作模式，本电路的 32 kHz 振荡都不会停止。

注意事项：

- *32 kHz 振荡电路没有振荡稳定等待功能或时钟故障检测功能。使能时钟协同功能，并可使用 CPU 域的时钟故障检测功能。*
- *放大器电路的供电电流随连接至 VBAT 域的振荡器的特性 (ESR) 和负载电容(CL) 的变化而变化。可对 VBAT 域和晶振器执行一项匹配评估，以此来确定适当的电流值。*
- *如果使用的 RTC 的 VBAT 域采用的是备用电源而非 VCC 电源，则应禁用与时钟发生器的协同操作功能。*
- *本电路的 32 kHz 振荡开始之后，不要更新 CCB 寄存器或 CCS 寄存器。*

32 kHz 振荡电路的应用

有关应用详细信息，参见“5 32 kHz 时钟设置步骤”。

32 kHz 振荡电路所用的寄存器

位	31 - 24	23 - 16	15 - 8	7 - 0	初始值	属性
	保留	保留	保留	VB_CLKDIV	0x00000007	R/W
	保留	保留	保留	WTOSCCNT	0x00000001	R/W
	保留	保留	保留	CCS	0x000000CE	R/W
	保留	保留	保留	CCB	0x000000CE	R/W
	保留	保留	保留	BOOST	0x00000003	R/W
	保留	保留	保留	EWKUP	0x00000000	R/W
	保留	保留	保留	VDET	0x00000080	R/W
	保留	保留	保留	HIBRST	0x00000000	R/W

上表所列寄存器的接口电路类型有 TYPE 1、TYPE 3 和 TYPE 4。

32 kHz 振荡电路的保存操作和回调操作分别为 PWRITE 和 PREAD。

2.4 上电电路

FM0+家族带有上电电路，该电路与检测 VBAT 域的上电情况的 VCC 电源引脚相互独立。

上电电路功能概述

VBAT 域中的上电电路具有以下功能：

VBAT 电源引脚上升沿检测功能

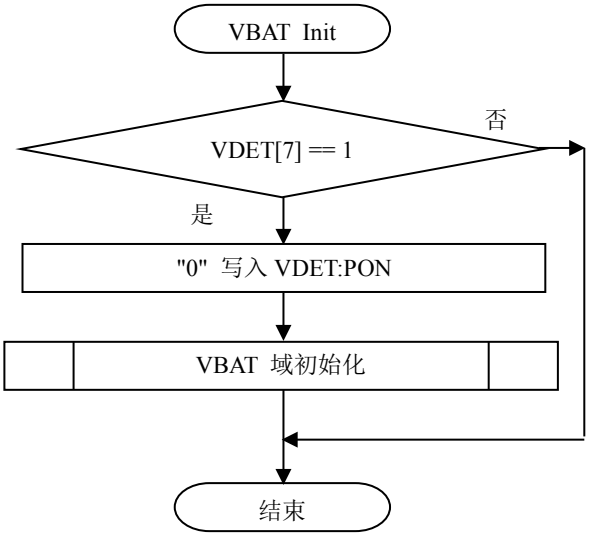
当上电电路输出上电信号时，VDET 寄存器的 bit7 读为 “1”。

上电信号保持生效，直至 VDET 寄存器的 bit7 写入 “0” 为止。

上电信号和 VDET 寄存器 bit7 的值不受 VCC 电源开断的影响。

当外设功能在 VCC 上升沿时被初始化时，如果按照下面的流程对 VBAT 域内的上电电路进行初始化，则正在运行的 VBAT 域初始化会被跳过，而 RTC 继续运行。

Figure 2-6 上电判断和 VBAT 电源域初始化流程



注意事项：

- *VBAT 域中的上电电路不带 VBAT 电源引脚低压检测功能。（FM0+ 家族不带 VBAT 电源引脚低压检测功能）。*
- *当上电信号保持生效时，VBAT 域内寄存器的值（RTC、32 kHz 振荡电路、VBAT I/O 端口控制电路、备用寄存器）保持各自的初始值不变。在设置这些电路之前，应在 VDET 寄存器的 bit7 写入“0”，清除上电信号。*

2.5 备用寄存器

FM0+家族配有 32 字节的备用寄存器，当 VBAT 电源通电时备用寄存器可以保持数据。

备用寄存器功能概述

向 VBAT 电源引脚供电时，备用寄存器会保持写入该寄存器的值。

VBAT 电源接通后，上电电路立即使备用寄存器复位。

VBAT 电源的使用情况可通过 VDET 寄存器的 bit7 位的读取值来检查，必要时应通过程序将备用寄存器初始化。

备用寄存器的配置和访问方法

本备用寄存器的接口电路类型为 TYPE 3。

详细信息参见“2.1 与常开域的接口”。

备用寄存器和数据保留寄存器之间的数据传输是对所有区的数据进行的批量传输。

数据更新步骤如下：

1. 设置 VB_CLKDIV 寄存器的值，使得传输时钟频率为 1 MHz 或更低。
2. 将数据保留寄存器中的数据回调到缓冲寄存器 (BREAD) 。

   当 WTCR20 寄存器的 bit2 写入 “1”时，开始执行回调操作，此时 WTCR10 寄存器的 bit7 读为“1”。当回调操作结束时，WTCR10 寄存器的 bit7 读取值变为“0”。
3. 修改缓冲寄存器的内容。

   缓冲寄存器允许执行随机读取访问和随机写入访问。
4. 将缓冲寄存器中的数据保存到数据保留寄存器中。

   当 WTCR20 寄存器的 bit3 写入 “1”时，启动保存操作，此时 WTCR10 寄存器的 bit7 读为“1”。当保存操作 (BWRITE)结束时，bit7 位变为 “0”。

*如果常开域的电源断开，缓冲寄存器内的数据会丢失。因此，应随时执行保存操作 (BWRITE) 将缓冲寄存器中保留的数据传输到数据保留寄存器中。

*当正在执行将数据传输到数据保留寄存器中时（WTCR10 寄存器的 bit7 读取值为“1”时），不要访问缓冲寄存器。

*如果在数据传输过程中常开域被复位或者 VCC 电源被切断，则不能保证数据保留寄存器中数据的完整性。

备用寄存器详细信息

■ 备用寄存器列表

位	31 – 24	23 – 16	15 – 8	7 – 0	初始值	属性
	BREG03	BREG02	BREG01	BREG00	0x00000000	R/W
	BREG07	BREG06	BREG05	BREG04	0x00000000	R/W
	BREG0B	BREG0A	BREG09	BREG08	0x00000000	R/W
	BREG0F	BREG0E	BREG0D	BREG0C	0x00000000	R/W
	BREG13	BREG12	BREG11	BREG10	0x00000000	R/W
	BREG17	BREG16	BREG15	BREG14	0x00000000	R/W
	BREG1B	BREG1A	BREG19	BREG18	0x00000000	R/W
	BREG1F	BREG1E	BREG1D	BREG1C	0x00000000	R/W

上述寄存器适用的接口电路类型为 TYPE 3。

备用寄存器的保存操作和回调操作分别为 BWRITE 和 BREAD。

备用寄存器通过 VBAT 引脚电源保持数据。

备用寄存器支持字节访问、半字访问和字访问。

2.6 VBAT I/O 端口

FM0+家族的 VBAT 域分配有四个 I/O 端口。这些 I/O 端口（VBAT I/O 端口）由 VBAT 域的端口控制电路（VBAT 端口控制电路）控制，即使在 VCC 电源断开时也可以继续运行。VBAT 端口控制电路与《FM0+家族外设手册》的“I/O 端口”一章中描述的端口控制电路是相互独立的。VBAT 端口控制电路的寄存器的地址被映射到不同于端口控制电路寄存器映射地址的区域。

VBAT I/O 端口功能概述

只要 VBAT 电源是接通的，即使 VCC 电源被切断，VBAT I/O 端口也会保持工作。

VBAT I/O 端口不具有重定位功能。

VBAT I/O 端口配置

VBAT 端口控制电路的寄存器用于选择 I/O 方向以及 GPIO 端口和外设功能 I/O 引脚之间的 I/O 端口的功能。

有关 VBAT I/O 端口的配置，参见“I/O 端口”一章中的 Figure 2-1。（对于具有相同功能的寄存器，替换为 Figure 2-1 中所用实际寄存器的名称。）

Table 2-8 所示为寄存器列表和各个寄存器的功能。

Table 2-8 寄存器列表和寄存器功能

寄存器名称	功能
VBPFR[5:4]	本寄存器用于设置 VBAT I/O 端口是用作专用引脚（用于振荡）还是数字 I/O 引脚。
VBPFR[3:0]	本寄存器设置 VBAT I/O 端口时用作 GPIO 端口还是外设功能 I/O 引脚。
VBPCR[3:0]	当 VBAT I/O 端口用作数字输入引脚或数字双向引脚时，可通过本寄存器设置 VBAT I/O 断开的上拉电阻的连接或断开。
VBDDR[3:0]	当 VBAT I/O 端口用作 GPIO 端口时，可通过本寄存器设置 GPIO 端口是用作输入引脚还是输出引脚。 Note: 当 VBAT I/O 端口用作外设功能 I/O 引脚时，本寄存器的设置无效。
VBDIR[3:0]	本寄存器用于读取 VBAT I/O 端口的电平。 – 当 VBAT I/O 端口用作数字输入引脚时，本寄存器读取输入电平。 – 当 VBAT I/O 端口用作数字输出引脚时，本寄存器读取输出电平。 – 当 VBAT I/O 端口用作数字专用引脚时，本寄存器读取值总为“0”。
VBDOR[3:0]	当 VBAT I/O 端口用作 GPIO 输出引脚时，本寄存器用于设置输出电平。 – 当 VBDOR[3:0]的位设置为“0”时，GPIO 输出引脚输出 “L” 电平。 – 当 VBDOR[3:0]的位设置为“1”时，GPIO 输出引脚输出“H” 电平。 注意：当 VBAT I/O 端口用作 GPIO 输入引脚或外设功能 I/O 引脚时，本寄存器的设置无效。
VBPZR[1:0]	本寄存器控制 VBAT I/O 端口的开漏。 – 当 VBAT I/O 端口输出“L”电平时，I/O 端口设置为 “L”电平输出。（不管 PCR 寄存器如何设置，上拉电阻都断开。） – 当 VBAT I/O 端口输出 “H” 电平时，I/O 端口设置为 “Hi-Z”，端口开漏采用伪开漏的方式进行控制。（不管 PCR 寄存器如何设置，上拉电阻都断开。） – 当 VBAT I/O 端口用作输入端口时，I/O 端口设置为“Hi-Z”， I/O 方向变为输入方向。（不管 PCR 寄存器如何设置，上拉电阻都断开。）

VBDIR 寄存器的接口类型为 TYPE 4，其它寄存器的接口电路类型为 TYPE 3。

32 kHz 振荡电路的保存操作和回调操作分别为 PWRITE 和 PREAD。（寄存器接口电路类型为 TYPE 3）

注意事项：

- *I/O 端口控制寄存器的设置（PFR4[6:9]、PCR4[6:9]、DDR4[6:9]、DIR4[6:9]、DOR4[6:9]、PZR4[6:9]）对 VBAT I/O 端口的操作无效。*

VBAT I/O 端口的初始设置

Table 2-9 列出了 VBAT I/O 端口的初始设置。

Table 2-9 VBAT I/O 端口初始设置

No.	引脚	初选功能
1	P46/X0A	本引脚可用作振荡引脚。（振荡已停止。） 数字输入已切断且已向本引脚输入“0”。
2	P47/X1A	本引脚可用作振荡引脚。（振荡已停止。） 数字输入已切断且已向本引脚输入“0”。
3	P48/VREGCTL	本引脚为数字输入引脚。输出为开漏。
4	P49/VWAKEUP	本引脚为数字输入引脚。输出为开漏。

当 VBAT 上电电路复位 VBAT 域时，VBATI/O 端口会分别保持 Table 2-9 所述状态。

VBAT I/O 端口的设置步骤

- 使用 32 kHz 振荡电路时

  设置步骤参见“5 32 kHz 时钟设置步骤”。

- 睡眠控制时

  睡眠控制设置步骤及 I/O 设置步骤参见 “4 睡眠控制” 。

- VBAT I/O 端口用作 GPIO 端口时

  设置方法参见《FM0+家族外设手册》的“I/O 端口”一章。

  （对于具有相同功能的寄存器，替换为该章中所用实际寄存器的名称。）

VBAT I/O 端口寄存器

■ VBAT I/O 端口寄存器列表

位	31 – 24	23 – 16	15 – 8	7 – 0	初始值	属性
	保留	保留	保留	VBPFR	0x0000001C	R/W
	保留	保留	保留	VBPCR	0x00000000	R/W
	保留	保留	保留	VBDDR	0x00000000	R/W
	保留	保留	保留	VBDIR	0x000000XX	R
	保留	保留	保留	VBDOR	0x0000000F	R/W
	保留	保留	保留	VBPZR	0x00000003	R/W

VBAT I/O 端口寄存器的配置及访问方法

VBAT I/O 端口寄存器所用的接口电路为 TYPE 3。

详细信息参见“2.1 与常开域的接口”。

缓冲寄存器和 VBAT I/O 端口寄存器之间的数据传输是对所有区的数据进行的批量传输。

数据更新步骤如下：

1. 设置 VB_CLKDIV 寄存器的值，使得传输时钟频率为 1 MHz 或更低。
2. 将数据从 VBAT I/O 端口保留寄存器回调到缓冲寄存器。

   当控制寄存器 20(WTCR20)的 VBAT 端口回调控制位(PREAD) 写入“1”时，回调操作启动，此时控制寄存器 10(WTCR10)的传输标志位(TRANS)变为“1”。
   当回调操作结束时，TRANS 位变为“0”。
3. 修改缓冲寄存器的内容。

   缓冲寄存器允许执行随机读取访问和随机写入访问。
4. 将缓冲寄存器中的数据保存到 VBAT I/O 端口寄存器中。

   当控制寄存器 20(WTCR20)内的 VBAT 端口保存控制位(PWRITE) 写入“1”时，保存操作启动，此时控制寄存器(WTCR10)的传输标志位(TRANS)变为“1”。
   当保存操作结束时，TRANS 位变为“0”。

- 单独修改缓冲寄存器内的新数据不会改变 VBAT I/O 端口引脚的状态。

  要修改 VBAT I/O 端口寄存器的值（引脚状态），需要执行保存操作，将与 VBAT I/O 端口寄存器相对应的缓冲寄存器中的数据传输到该 VBAT I/O 端口寄存器。
- 将数据传输到数据保留寄存器（WTCR10 寄存器的 TRANS 读值为“1”）时，不要访问缓冲寄存器。
- 如果在数据传输过程中常开域被复位或者 VCC 电源被切断，则不能保证数据保留寄存器中数据的完整性。

3. 芯片电源控制

本节说明芯片电源的应用和切断

VCC 电源和 VBAT 电源组合表

Table 3-1 列出了 VCC 电源和 VBAT 电源各自的状态。

Table 3-1VCC 电源状态和 VBAT 电源状态的组合

	VBAT 电源开	VBAT 电源关闭
VCC 电源开	正常运行	禁止此项组合。
VCC 电源关	只有 VBAT 域继续工作。	停止运行

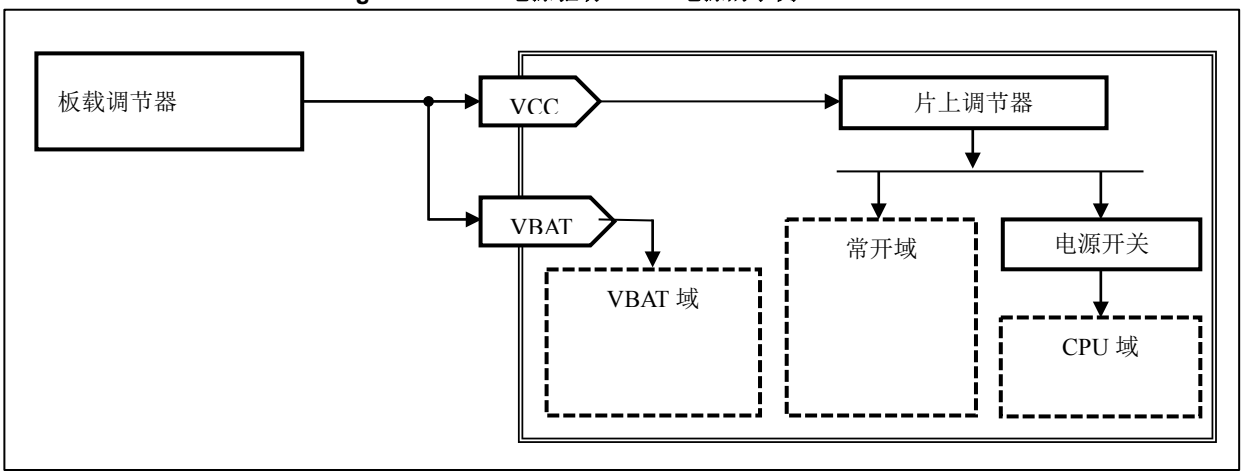
VBAT 电源和 VCC 电源由同一电源驱动

■ 电源状态的转换

当 VBAT 电源和 VCC 电源由同一电源驱动时，芯片电源状态如 Table 3-1 所示在“正常运行”和“停止运行”之间转换。

当同一电源驱动 VBAT 电源和 VCC 电源时，每次使用 VCC 电源时都应对 VBAT 域进行初始化。

Figure 3-1VCC 电源驱动 VBAT 电源的示例



如果 VBAT 电源不采用备用电源，应将 VBAT 电源引脚直接连接到芯片内的 VCC 电源引脚。

当 VBAT 电源引脚直接连接至 VCC 电源引脚时，可以避免禁用组合“VCC 电源开及 VBAT 电源关”。

电池驱动 VBAT 电源

■ 电源状态的转换

Figure 3-2 所示为 VBAT 电源采用电池时的电源状态转换。Figure 3-3 所示为各个电路对应的波形图。

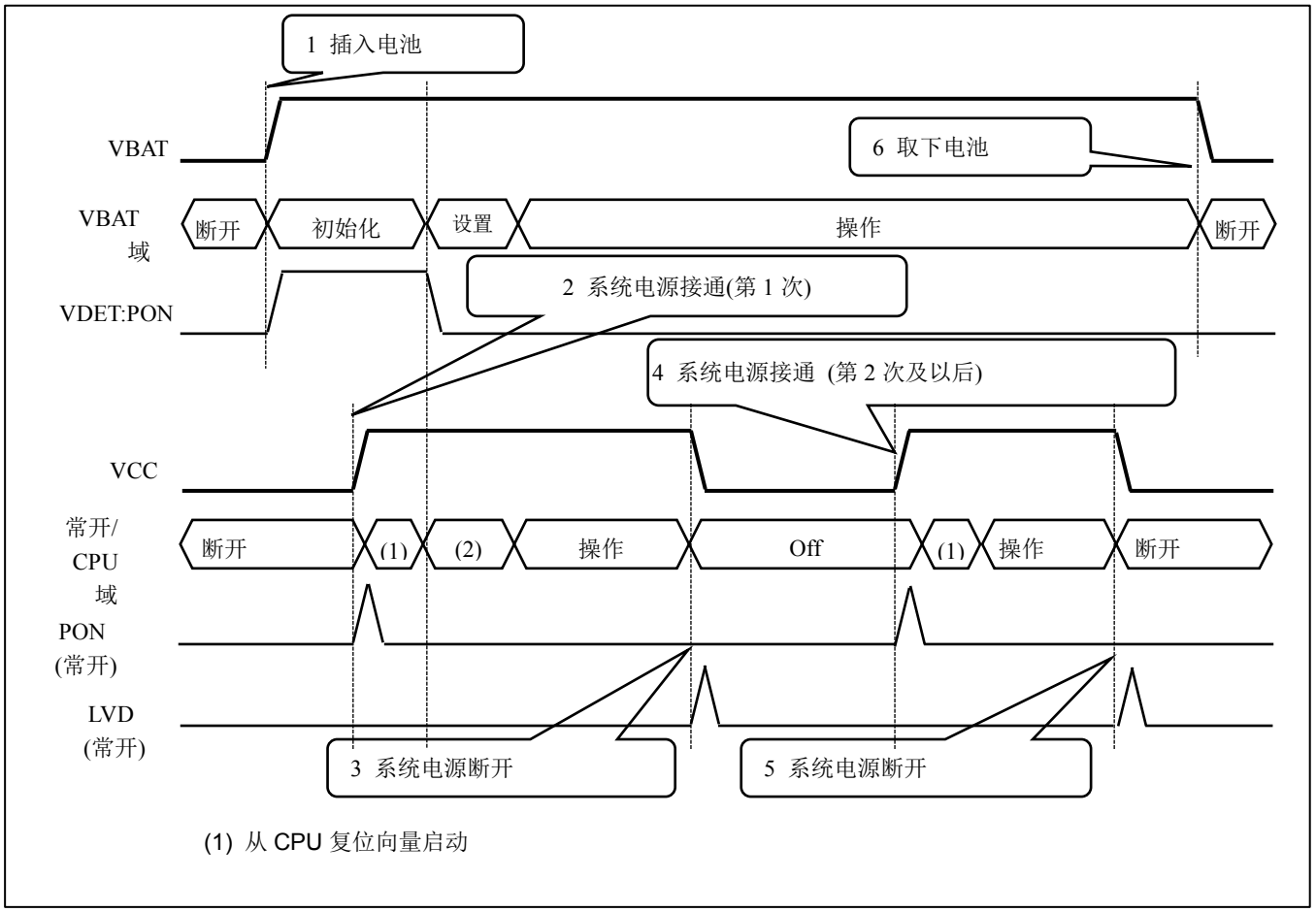
VDET 寄存器的 Bit7 用于指示系统电源是否在第一次接通。

如果系统电源第一次未被接通，应对 VBAT 域的电路进行设置。

Figure 3-2 VBAT 电源采用电池时的状态转换



Figure 3-3 采用电池时的波形示例



(1) 从 CPU 复位向量启动

(2) VBAT 域的设置

■ 电源配置示例

Figure 3-4 原电池用作备用电源的示例

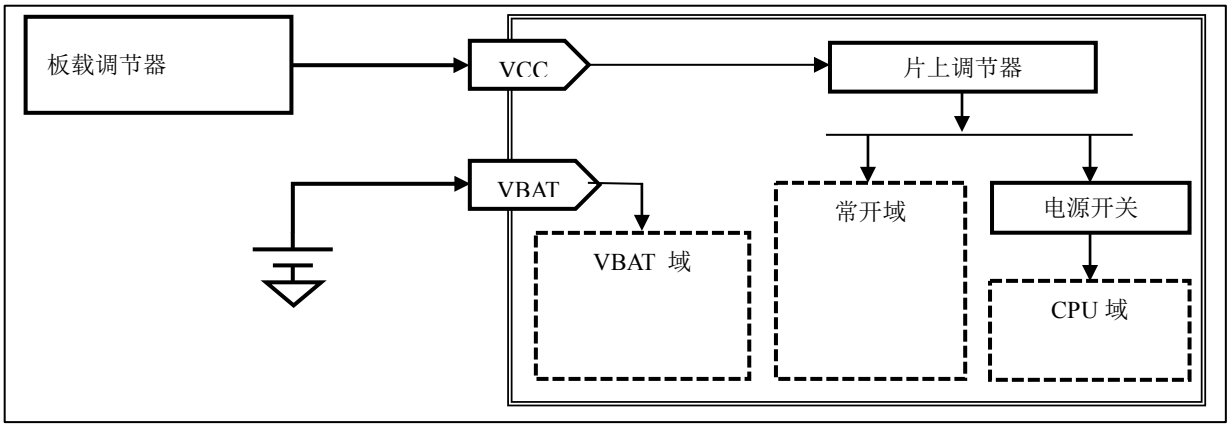
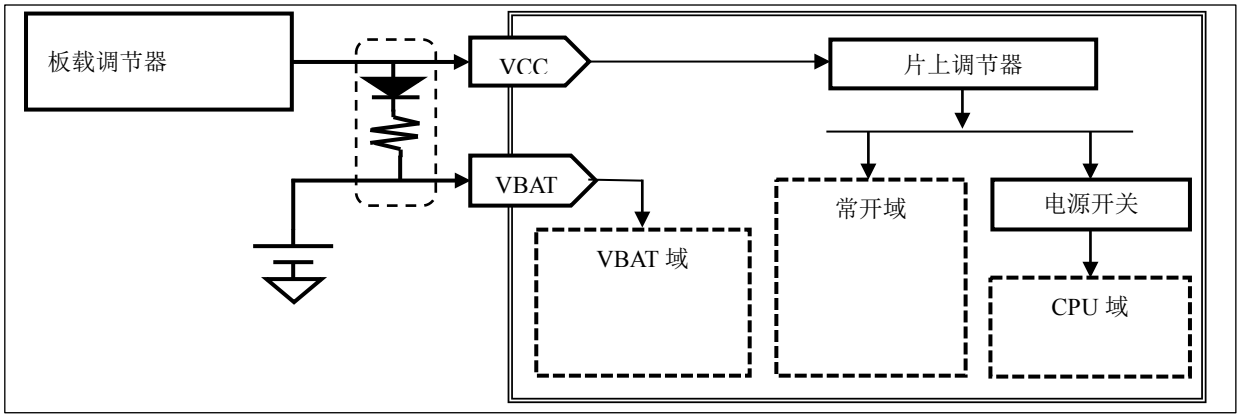


Figure 3-5 蓄电池用作备用电池的示例



虚线框内的二极管和电阻器对蓄电池进行涓流充电。

设置电阻时，应检查所用的蓄电池能否进行涓流充电，并检查最大的涓流充电电流。

如果所用的蓄电池不能进行涓流充电，应将其作为原电池使用。

注意事项：

- *在装入或取下电池前请断开系统电源。*
- *如果备用电源采用原电池，则不建议用二极管连接系统电源至备用电源。*

4. 睡眠控制

本节以实例说明通过微控制器进行片外电源门控的电路配置和序列。

睡眠控制概述

通过 VBAT 域控制板载调节器的待机功能，睡眠控制对 VCC 电源（常开域和 CPU 域）的开和关进行控制。

要执行睡眠控制时，使用备用电源而非 VCC 电源（系统电源）向 VBAT 引脚供电。

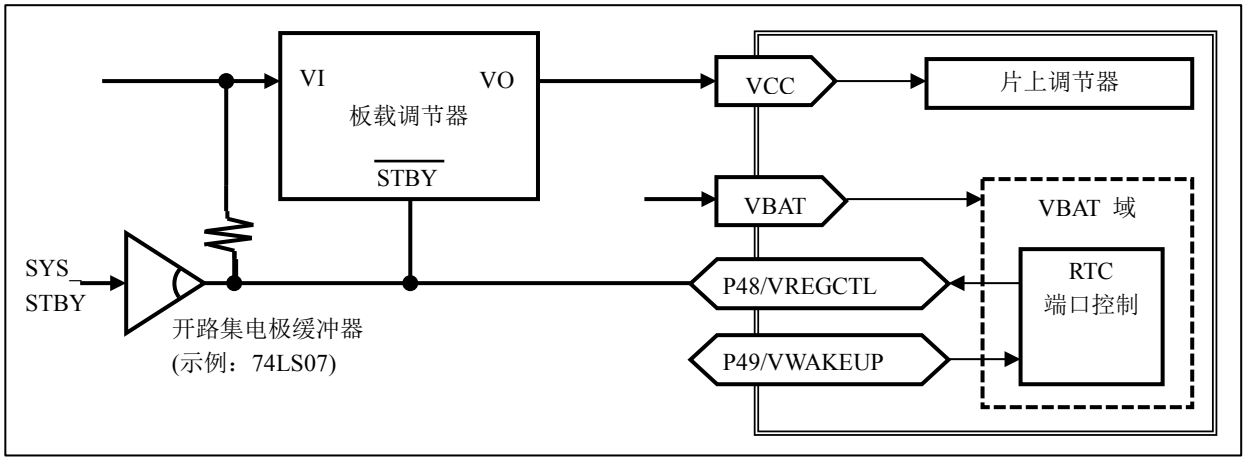
以下为使微处理器从睡眠状态唤醒的两种唤醒源：

- RTC 报警中断
- 对 P49/VWAKEUP 引脚的唤醒请求（在上升沿发出的请求）。

如需采用 RTC 报警中断唤醒源将微处理器从睡眠状态唤醒，即使 VCC 电源断开也要保持 32 kHz 振荡工作。

FM0+家族产品的外部连接示例

Figure 4-1 板载调节器的输入电压(VI) 低于 5.5 V 时进行外部连接的示例



许多板载调节器都会在 “STBY”输入为“L” 电平时进入待机状态。

VBAT I/O 端口的 P48/VREGCTL 引脚为耐 5 V 伪开漏引脚。

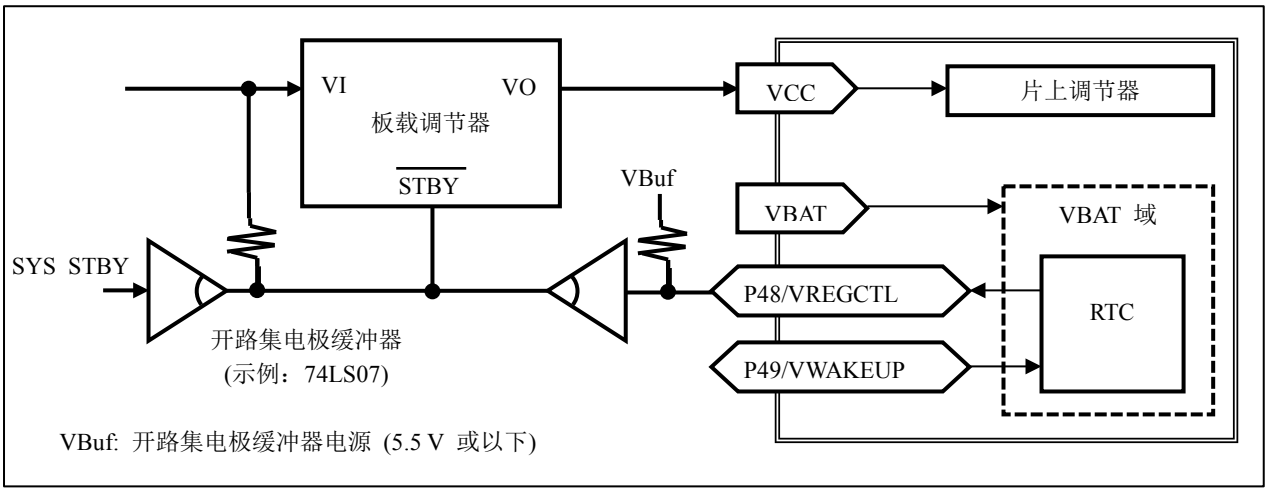
将 P48/VREGCTL 连接至板载调节器的“STBY”输入端，再将上拉电阻器连接至板载调节器的输入电压 (VI) 。

Table 4-1 列出系统待机信号被集电极开路缓冲器缓冲且缓冲的待机信号和 P48/VREGCTL 通过有线“或”逻辑电路连接时，板载调节器是如何工作的。

Table 4-1 板载调节器的工作情况

SYS_STBY	VREGCTL =“L”	VREGCTL =“H”
“L”	待机模式	待机模式
“H”	待机模式	正常工作模式

Figure 4-2 板载调节器的输入电压(VI) 高于 5.5 V 时进行外部连接的示例

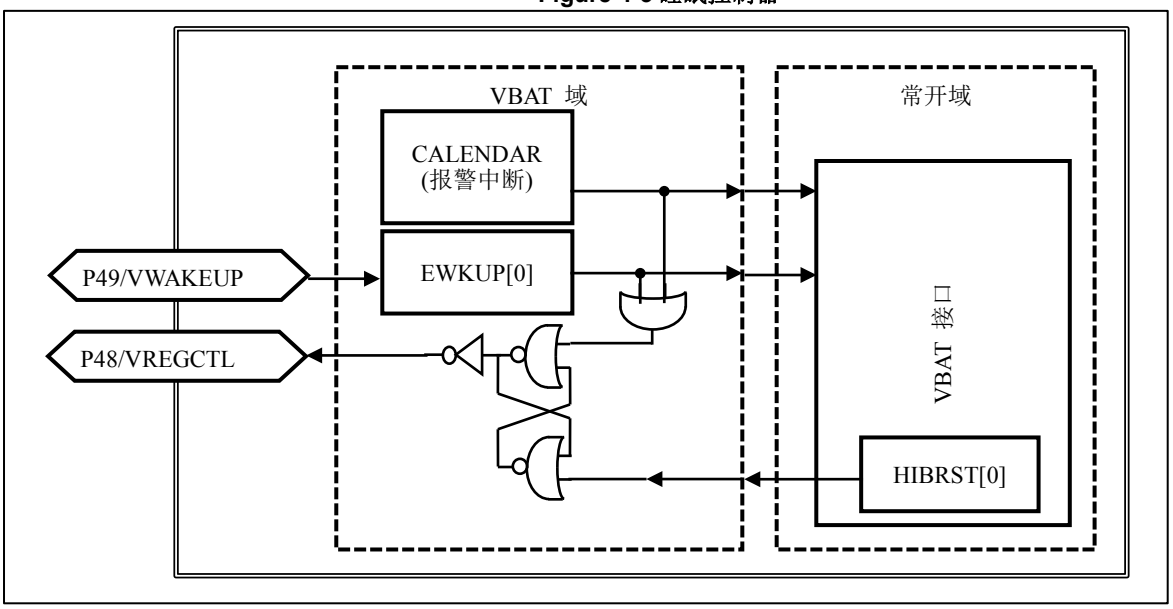


当板载调节器的输入电压高于 5.5 V 时，板载调节器的待机引脚不能直接通过 P48/VREGCTL 引脚控制。应采用耐压性高于板载调节器输入电压的开路集电极缓冲器进行缓冲。

睡眠控制器配置框图

睡眠控制器是 RTC 电路的组成部分。Figure 4-3 所示为睡眠控制器的配置。

Figure 4-3 睡眠控制器



睡眠操作流程示例

Figure 4-4 所示为睡眠操作流程

Figure 4-4 睡眠操作流程示例



■ 睡眠操作初始设置

以下所述为睡眠操作所需的初始设置：

- RTC 的报警设置

  报警设置的方法参见《FM0+家族外设手册》“计时器部分”的“RTC 计数模块”一章。

- P49/VWAKEUP 引脚的设置

  在 VBPFR 寄存器的 VPFR1bit 写入 “1”。

- P48/VREGCTL 引脚的设置

  在 VBPFR 寄存器的 VPFR0bit 写入 “1”。

即使 RTC 的报警设置和 P49/WAKEUP 引脚设置都未进行，CPU 内核也可以切换到睡眠状态。

如果 CPU 内核在上述两项设置都未进行的情况下切换到睡眠状态，则无法唤醒至正常工作状态。

■ 睡眠启动的设置

RTC 报警中断和唤醒(P49/VWAKEUP 引脚)都被清除时，如果 HIBRST 寄存器的 bit0 写入“1”，则 P48/VREGCTL 引脚会变为“0”，板载调节器转换至待机状态，且 VCC 电源断开。

■ 如何判断板载调节器是否从睡眠状态唤醒以及唤醒后的操作

当发生 RTC 报警中断或唤醒请求时，P48/VREGCTL 引脚会变为 “1”，板载调节器会从待机状态唤醒，且 VCC 电源接通。

当 VCC 电源接通时，CPU 内核执行正常的上电操作。

判断 CPU 内核是否从睡眠状态唤醒的方法是检查下列三个条件是否满足：

- VBAT 域已上电(VDET[7])。
- 已发生 RTC 报警中断（WTCR12 寄存器）。
- 已发出唤醒请求(EWKUP[0])。

注意事项：

- *当 HIBRST 寄存器的 bit0 被写入 “1” 时，P48/VREGCTL 引脚立即变为“0” 。*
- *在 HIBRST 寄存器的 bit0 位写入 “1” 之前应完成切断 VCC 电源的所有操作。*
- *在睡眠状态下进行操作时，VCC 电源被假定为切断，由 P48/VREGCTL 引脚控制。*
  在调试时如果板载调节器不是直接由 P48/VREGCTL 引脚控制，应通过手动操作切断 VCC 电源。

5. 32 kHz 时钟设置步骤

本节说明使用 RTC 时设置 32 kHz 振荡电路的建议序列。

VBAT 域中 32 kHz 振荡电路的特性

当 VBAT 域中集成了 32 kHz 振荡电路时，即使 CPU 域和常开域电源被断开，32 kHz 振荡电源仍然可以继续工作，RTC 也可以继续计时。

与时钟控制电路连接

如果 VBAT 域中的 32 kHz 振荡电路与时钟控制电路连接，则 VBAT 域可与 FM3 产品族兼容。

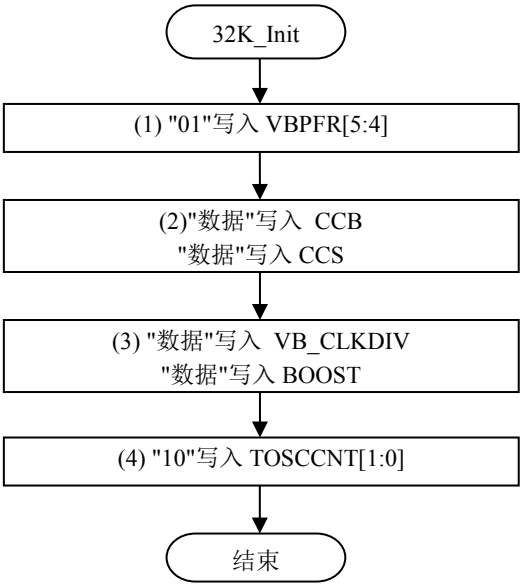
此外，VBAT 域中的 32 kHz 振荡电路与时钟控制电路连接而 VCC 电源断开，此时如果 VCC 电源开启时 CPU 切换到深度待机模式或深度待机停止模式，32 kHz 振荡电路会自动停止工作。

如果 32 kHz 时钟只用作子操作模式下的时钟，可将 32 kHz 振荡电路与时钟控制电路连接，降低备用电源的功耗。

■ 设置步骤示例

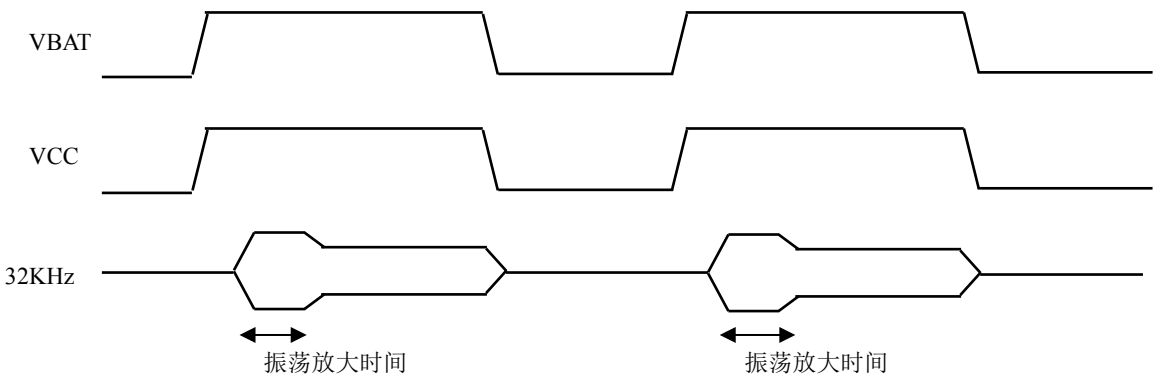
(1) 将 VBAT I/O 端口设置为使用 32KHz 振荡电路。
(2) 流向振荡放大器电路的电流设置恰当的电流值。
(3) 设置振荡增压时间。
(4) 使能与时钟控制电路的协同操作。

此外，使能振荡功能。

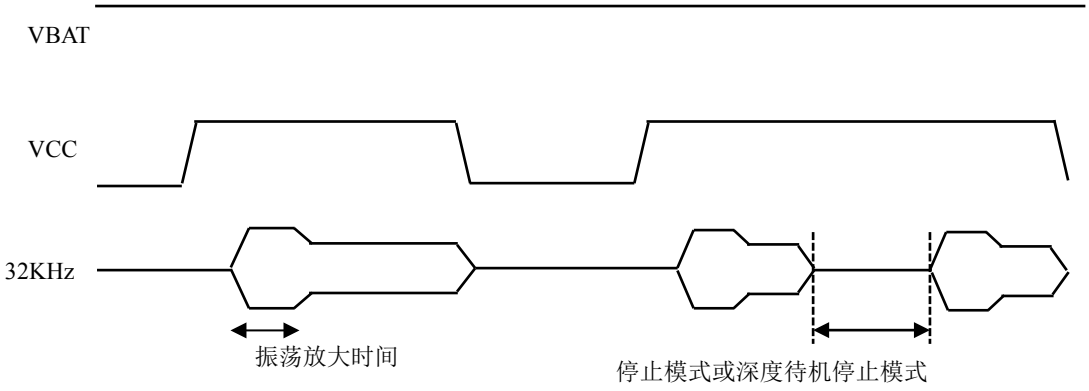


■ 操作示例

– 不使用备用电源。



– 使用备用电源，且 32 kHz 振荡电路与时钟控制电路连接。



不与时钟控制电路连接

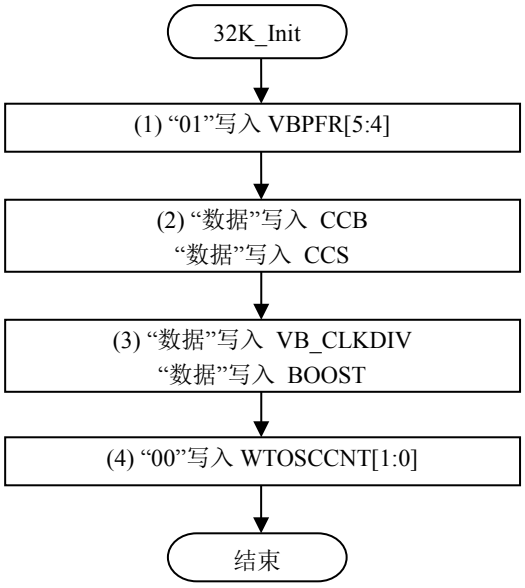
如果总是使用备用电源保持 RTC 工作，则不要将 32 kHz 振荡电路与时钟控制电路连接。

执行以下操作可以降低整个系统的平均功耗：备用电源只用于保持 VBAT 运行，当 CPU 不需要进行处理时采用 VBAT 域的睡眠控制功能或外部电路切断 VCC 电源。

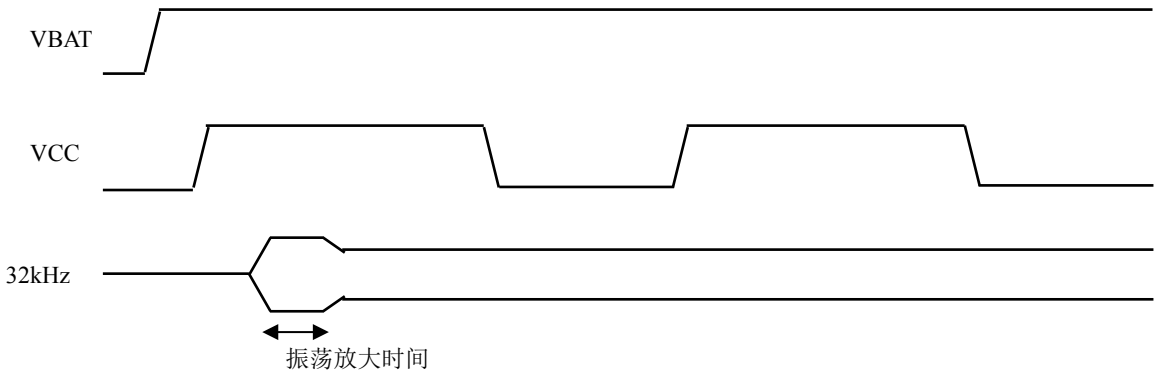
■ 设置步骤示例

(1) 将 VBAT I/O 端口寄存器设置为使用 32 kHz 振荡电路。
(2) 流向振荡放大器电路的电流设置恰当的电流值。
(3) 设置振荡增压时间。
(4) 禁用与时钟控制电路协同操作。

此外，使能振荡功能。



■ 操作示例



不与时钟控制电路连接，但要等待振荡稳定

如果始终用备用电源保持 RTC 工作，则不必要将 32 kHz 振荡电路与时钟控制电路连接。但是，32 kHz 振荡电路和 VBAT 域中的 RTC 不具有振荡稳定等待功能。

当 32 kHz 时钟只用于 RTC 时，可采用软件计时器对振荡稳定等待时间进行计时。不过，如果 32 kHz 时钟也在子操作模式下使用，就需要使用时钟振荡稳定等待功能。

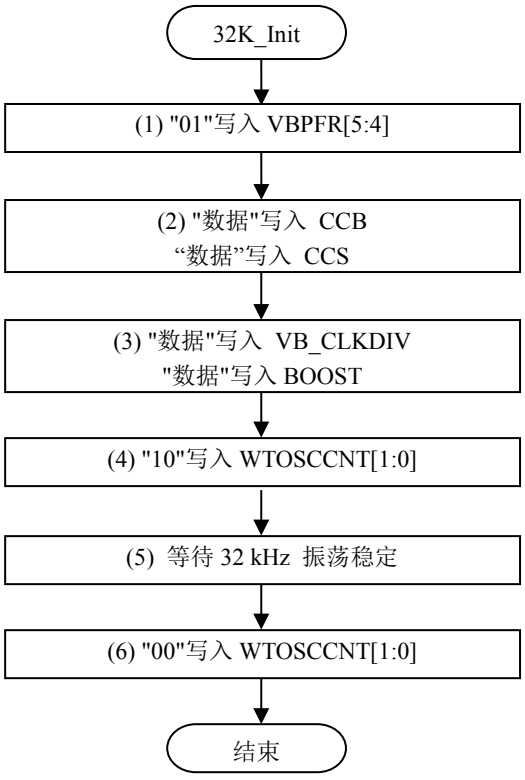
在上述情形下，只能在振荡开始时按照下列步骤使能振荡稳定等待功能。

■ 设置步骤示例

使能与时钟控制电路的协同操作，启动振荡。

当振荡稳定等待时间过去后，禁用与时钟控制电路的协同操作。

(1) 将 VBAT I/O 端口寄存器设置为使用 32 kHz 振荡电路。
(2) 流向振荡放大器电路的电流设置恰当的电流值。
(3) 设置振荡增压时间。
(4) 使能与时钟控制电路的协同操作。此外，使能振荡功能。
(5) 等待 32 kHz 振荡稳定。
(6) 禁用与时钟控制电路的协同操作。（振荡保持使能状态。）



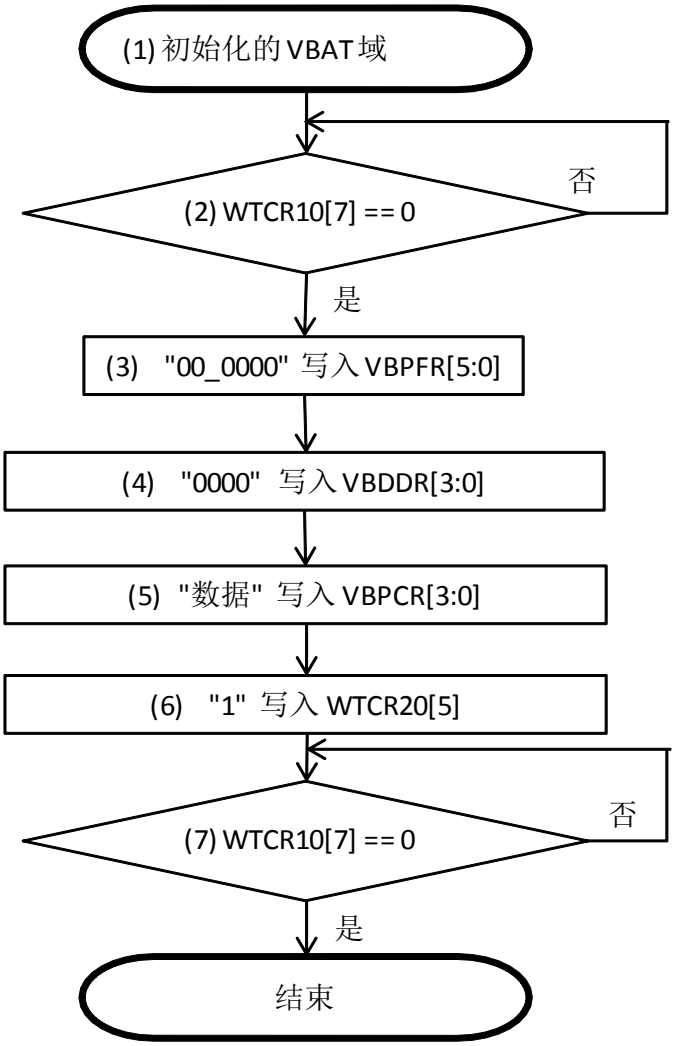
6. VBAT I/O 端口设置步骤

■ 当 VBAT I/O 端口用作通用 I/O 输入端口时

设置步骤示例

以下是 P46、P47、P48 和 P49 全部用作通用 I/O 输入端口时的设置示例。

(1) 启动 VBAT 域（参见 Figure 2-6）。
(2) 将通用 IO 端口设置为 GPIO 引脚。
(3) 上拉设置。
(4) 将端口输出方向设置为输入方向。
(5) 将设定值传输到 VBAT 域。
(6) 等待传输完成。

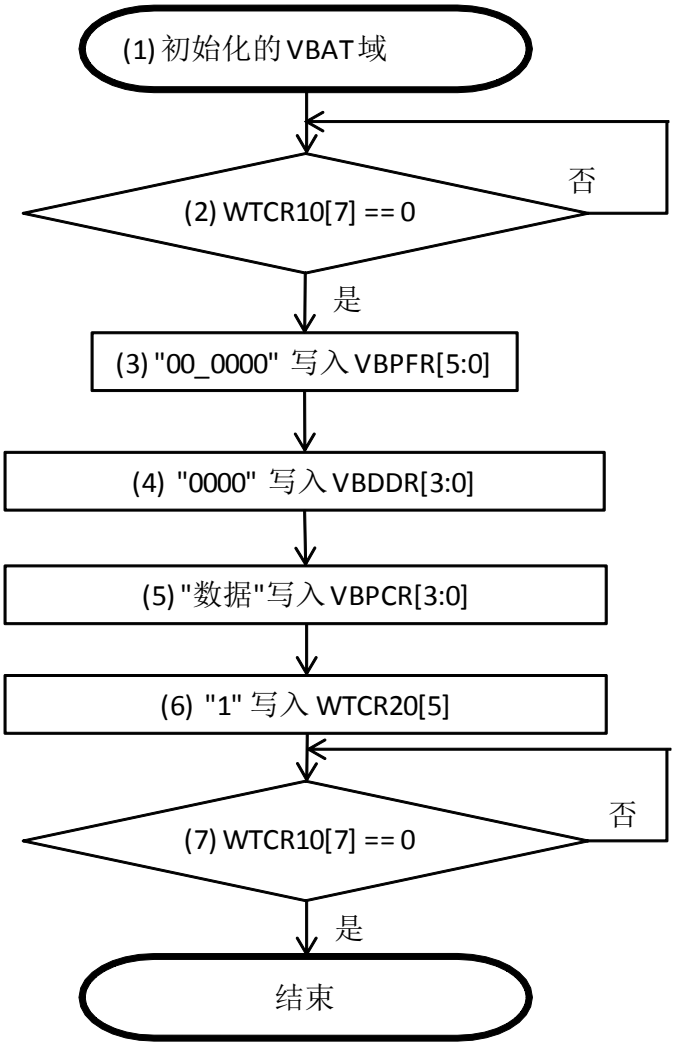


■ 当 VBAT I/O 端口用作通用 I/O 输出时

设置步骤示例

以下是 P46、P47、P48 和 P49 全部用作通用 I/O 输出时的设置示例。

(1) 启动 VBAT 域（参见 Figure 2-6）。
(2) 将通用 IO 端口设置为 GPIO 引脚。
(3) 将端口的输出方向。
(4) 设置端口输出数据寄存器。
(5) 设置端口伪开漏寄存器（只有 P48 和 P49 可设置）。
(6) 将设定值传输到 VBAT 域。
(7) 等待传输完成。

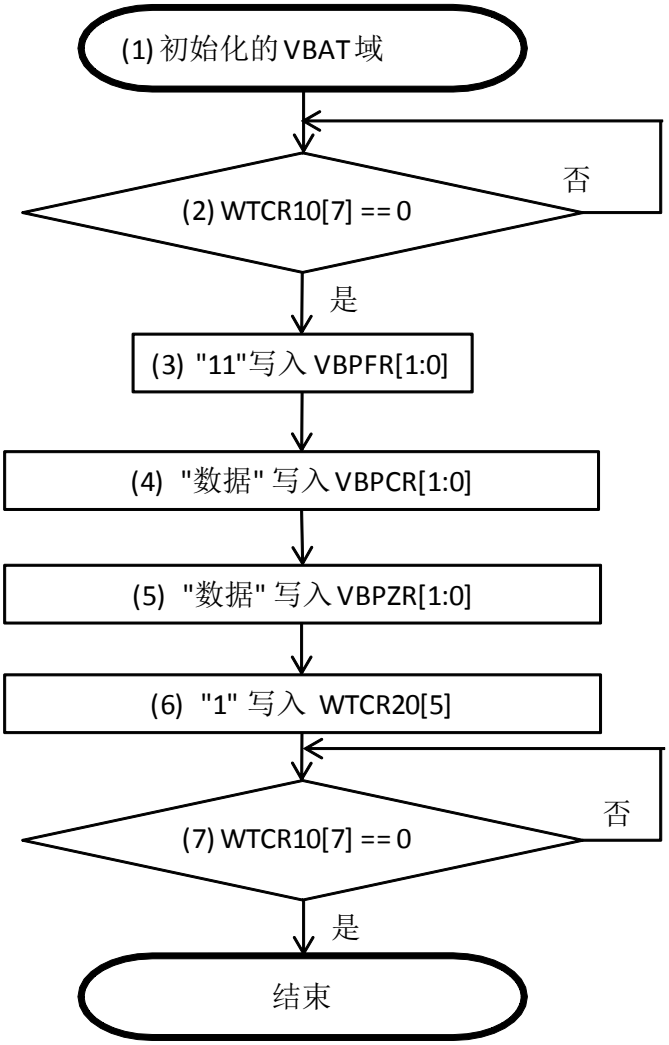


■ 当 VBAT I/O 用作外设功能时

设置步骤示例

以下是 P48 和 P49 用作外设功能时的设置示例：

(1) 启动 VBAT 域（参见 Figure 2-6）。
(2) 将通用 IO 端口设置为外设功能。
(3) 上拉设置。
(4) 设置端口伪开漏。
(5) 将设定值传输到 VBAT 域。
(6) 等待传输完成。



7. 寄存器

本节说明 VBAT 域单元内的寄存器列表。

Table 7-1 列出了 VBAT 域单元内的寄存器。

Table 7-1 VBAT 域单元寄存器

缩写	寄存器名称	参考章节
VB_CLKDIV	VB_CLKDIV 寄存器	7.1
WTOSCCNT	WTOSCCNT 寄存器	7.2
CCS/CCB	CCS/CCB 寄存器	7.3
BOOST	BOOST 寄存器	7.4
EWKUP	EWKUP 寄存器	7.5
HIBRST	HIBRST 寄存器	7.6
VDET	VEDT 寄存器	7.7
VBPFR	端口功能设置寄存器	7.8
VBPCR	上拉设置寄存器	7.9
VBDDR	端口 I/O 方向设置寄存器	7.10
VBDIR	端口 I/O 数据寄存器	7.11
VBDOR	端口输出数据寄存器	7.12
VBPZR	端口伪开漏设置寄存器	7.13

Table 7-1 所列常开域中的寄存器和缓冲器（除 VBDIR、VDET 和 EWKUP 外）会被系统复位或 RTC 复位清除。因此，在重新设置了值或执行了回调操作之后必须进行保存操作。

7.1 VB_CLKDIV 寄存器

VB_CLKDIV 寄存器用于在备用寄存器和端口寄存器同时被传输时设置传输时钟的频率。

位	7	6	5	4	3	2	1	0
字段	DIV7	DIV6	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	1	1	1

本寄存器所用的接口电路为 TYPE 1。

[bit7:0] DIV[7:0]：PREAD、PWRITE、BREAD、BWRITE 的传输时钟设置位

这些位用于设置备用寄存器和端口寄存器进行批量传输时所用的传输时钟周期。

寄存器值的计算公式：传输时钟 = PCLK / (VB_CLKDIV + 2)

（这些位的设置值应当使得 BREAD/BWRITE 和 PREAD/PWRITE 所用的传输时钟频率为 1 MHz 或更低）。

7.2 WTOSCCNT 寄存器

WTOSCCNT 寄存器用于指定 32 kHz 振荡电路。

位	7	6	5	4	3	2	1	0
字段	保留						SOSCNTL	SOSCEX
属性	-						R/W	R/W
初始值	-						0	1

本寄存器所用的接口电路为 TYPE 3。

[bit7:2]保留：保留位

这些位读作“0b000000”。

写入这些位时，写入 “0b000000” 。

[bit1] SOSCNTL：协同操作控制位

本位用于使能或禁用 32 kHz 振荡电路和时钟控制电路之间的协同操作。

位		说明
读取		进行读取访问时，读取本位的值。
写入	0	32 kHz 振荡电路独立用作 VBAT 域。(初始值)
	1	32 kHz 振荡电路与时钟控制电路连接。

[bit0] SOSCEX：振荡使能位

本位用于在 32 kHz 振荡电路独立用作 VBAT 时使能或禁用振荡电路。

如果 32 kHz 振荡电路与时钟控制电路连接，本位不能控制振荡电路的操作。

位		说明
读取		进行读取访问时，读取本位的值。
写入	0	启动振荡。
	1	停止振荡。(初始值)

7.3 CCS/CCB 寄存器

CCS 寄存器用于设置振荡持续进行时的电流值。
CCB 寄存器在振荡开始时设置增压电流。

TYPE 2-M0+产品

■CCS 寄存器

位	7	6	5	4	3	2	1	0
字段	CCS							
属性	R/W							
初始值	11001110							

本寄存器所用的接口电路为 TYPE 3。

[bit7:0] CCS：振荡持续电流设置位
本位用于设置持续振荡时的电流值。

■CCB 寄存器

位	7	6	5	4	3	2	1	0
字段	CCB							
属性	R/W							
初始值	11001110							

本寄存器所用的接口电路为 TYPE 1。

[bit7:0] CCB：振荡增压电流设置位
本位用于在振荡开始时设置增压电流值。

Table7-2CCS/CCB 寄存器设置和模式之间的关系

CCS/CCB	模式	备注
00000100	低功耗	装载能力方面可考虑使用较小型的晶振器。
11001110	标准	(初始值)
其它	未定义	禁止设置

注意事项：

- *CCS 寄存器和 CCB 寄存器采用同样的设置。*

7.4 BOOST 寄存器

BOOST 寄存器用于设置振荡增压的时钟值。

位	7	6	5	4	3	2	1	0
字段	保留						BOOST1	BOOST0
属性	-						R/W	R/W
初始值	-						1	1

本寄存器所用的接口电路为 TYPE 1。

[bit7:2]保留：保留位

这些位读作“0b000000”。

写入这些位时，写入 “0b000000” 。

[bit1:0] BOOST1、BOOST0：振荡增压时间设置位

本位用于设置振荡增压的时钟值。

Table 7-3 振荡增压时间设置

BOOST1	BOOST0	振荡增压时间
1	1	500 ms[初始值]
1	0	250 ms
0	1	125 ms
0	0	62.5 ms

7.5 EWKUP 寄存器

EWKUP 寄存器用于显示和清除唤醒的请求状态。

位	7	6	5	4	3	2	1	0
字段	保留							WUP0
属性	-							R/W
初始值	-							0

本寄存器所用的接口电路为 TYPE 4。

[bit7:1]保留：保留位

这些位读作 “0b0000000”。

写入这些位时，写入 “0b0000000” 。

[bit0] WUP0：唤醒请求位

位		说明
读取	0	VBAT 域尚未收到任何唤醒请求。
	1	VBAT 域已收到唤醒请求。
写入	0	VBAT 域清除唤醒请求。
	1	本位写入“1”时，操作无效。

当 VBPFR 寄存器的 VPFR0 设置为“1”时，唤醒请求功能使能。

使能唤醒请求功能后，如果 VBAT 域检测到 P49/VWAKEUP 引脚的上升沿，则 VBAT 域会接受一个唤醒请求，并使 RTC 中断，从而中断控制电路。

唤醒请求会在从唤醒请求清除后的第 7 个 PCLK 周期或之后被接收。清除 VBAT 域中的寄存器需要 7 个 PCLK 周期。因此，如果在 7 个 PCLK 周期过去之前进入了待机模式或输入了外部复位，则唤醒请求无法被接收，因为 PCLK 停止了工作。

7.6 HIBRST 寄存器

HIBRST 寄存器用于设置睡眠启动。

位	7	6	5	4	3	2	1	0
字段	保留							HIBRST
属性	-							R/W
初始值	-							0

本寄存器所用的接口电路为 TYPE 1。

[bit7:1]保留：保留位

这些位读作 “0b0000000”。

写入这些位时，写入“0b0000000”。

[bit0] HIBRST：睡眠请求位

位		说明
读取		读取本位的值。
写入	0	本位可写入“0”，但睡眠操作时写入无效。
	1	启动睡眠状态。

当端口功能设置寄存器(VBPFR)的 VPFR0 位设置为“1”时，启动睡眠状态。

7.7 VDET 寄存器

VDET 寄存器用于指示上电电路的状态，并清除上电信号。

位	7	6	5	4	3	2	1	0
字段	PON	保留						
属性	R/W	-						
初始值	1	-						

本寄存器所用的接口电路为 TYPE 4。

[bit7] PON：上电位

本位用于指示上电电路的状态并清除上电信号。

位		描述
读取	0	指示上电电路的初始化信号已被清除。
	1	指示上电电路已输出初始化信号。
写入	0	清除上电信号。
	1	本位写入“1”时操作无效。

[bit6:0]保留：保留位

这些位读作 “0b0000000”。

写入这些位时，写入“0b0000000” 。

7.8 端口功能设置寄存器 (VBPFR)

VBPFR 寄存器选择引脚的使用。

位	7	6	5	4	3	2	1	0
字段	保留		SPSR1	SPSR0	VPFR3	VPFR2	VPFR1	VPFR0
属性	-		R/W	R/W	R/W	R/W	R/W	R/W
初始值	-		0	1	1	1	0	0

本寄存器所用的接口电路为 TYPE 3。

[bit7:6]保留：保留位

这些位读作 “0b00”。

写入这些位时，写入 “0b00”。

[bit5:4] SPSR1, SPSR0: 振荡引脚功能设置位

bit5	bit4	功能
0	0	P46 和 P47 引脚用作数字(GPIO)引脚。
0	1	P46 和 P47 引脚用作 32 kHz 振荡引脚。[初始值]
1	0	P46 和 P47 引脚用作数字(GPIO)引脚。
1	1	P46 引脚用作外部时钟输入引脚。 P47 引脚用作数字(GPIO)引脚。

[bit3] VPFR3: P46/X0A 引脚端口功能设置位

[bit2] VPFR2: P47/X1A 引脚端口功能设置位

位		描述
读取		进行读取访问时，读取本位的值。
写入	0	VPFR3 位或 VPFR2 位对应的引脚被用作 GPIO 端口。
	1	VPFR3 位或 VPFR2 位对应的引脚被用作外设功能 I/O 引脚。(初始值)

[bit1] VPFR1：P49/VWAKEUP 引脚端口功能设置位

[bit0] VPFR0: P48/VREGCTL 引脚端口功能设置位

位		描述
读取		进行读取访问时，读取本位的值。
写入	0	VPFR1/VPFR0 位对应的引脚被用作 GPIO 端口。[初始值]
	1	VPFR1/VPFR0 位对应的引脚被用作外设功能 I/O 引脚。

VBPFR[5:2]设置组合参见 Table 7-4。

Table 7-4 VBPFR[5:2] 设置组合

	VBPFR[5]	VBPFR[4]	VBPFR[3]	VBPFR[2]
GPIO	0	0	0	0
32kHz 振荡	0	1	-	-
GPIO	1	0	0	0
P46 外部时钟输入	1	1	1	0

要使用 32kHz 振荡电路时，将振荡引脚的功能设置位 (VBPFR[5:4])设置为 0b01。这样可以使能 32kHz 振荡电路，无需依靠 VBPFR[3:2]。

要将 P46/X0A 用作外部时钟时，应将 VBPFR[5:2]设置为 0b1110，然后从 P46/X0A 输入外部时钟。在这种情况下，P47 可用作 GPIO 引脚。

7.9 上拉设置寄存器 (VBPCR)

VBPCR 寄存器用于设置引脚的上拉。.

TYPE 2-M0+产品没有 P46/X0A 引脚和 P47/X1A 引脚的上拉功能，因此 VBPCR[3:2]的设置无效。

位	7	6	5	4	3	2	1	0
字段	保留				VPCR3	VPCR2	VPCR1	VPCR0
属性	-				R/W	R/W	R/W	R/W
初始值	-				0	0	0	0

本寄存器所用的接口电路为 TYPE 3。

[bit7:4]保留：保留位

这些位读作 “0b0000”。

写入这些位时，写入“0b0000” 。

[bit3] VPCR3: P46/X0A 引脚上拉设置位

[bit2] VPCR2: P47/X1A 引脚上拉设置位

[bit1] VPCR1: P49/VWAKEUP 引脚上拉设置位

[bit0] VPCR0: P48/VREGCTL 引脚上拉设置位

位		描述
读取		进行读取访问时，读取本位的值。(初始值= 0)
写入	0	VPCR3/VPCR2/VPCR1/VPCR0 位对应引脚的上拉电阻器从该引脚断开。
	1	当 VPCR3/VPCR2/VPCR1/VPCR0 位对应的引脚处于输入状态（GPIO 功能或外设功能）时，上拉电阻器连接至引脚。 当 VPCR3/VPCR2/VPCR1/VPCR0 位对应的引脚处于输出状态时，上拉电阻器与引脚断开。

7.10 端口 I/O 方向设置寄存器 (VBDDR)

VBDDR 寄存器用于设置引脚的 I/O 方向。

TYPE 2-M0+产品中，P46/X0A 引脚和 P47/X1A 引脚的 GPIO 功能只是输入，因此这些端口不能用作输出端口。

位	7	6	5	4	3	2	1	0
字段	保留				VDDR3	VDDR2	VDDR1	VDDR0
属性	-				R/W	R/W	R/W	R/W
初始值	-				0	0	0	0

本寄存器所用的接口电路为 TYPE 3。

[bit7:4]保留：保留位

这些位读作 “0b0000”。

写入这些位时，写入“0b0000”。

[bit3] VDDR3: P46/X0A 引脚端口方向设置位

[bit2] VDDR2: P47/X1A 引脚端口方向设置位

[bit1] VDDR1: P49/VWAKEUP 引脚端口方向设置位

[bit0] VDDR0: P48/VREGCTL 引脚端口方向设置位

位		描述
读取		进行读取访问时，读取本位的值。(初始值= 0)
写入	0	GPIO 端口用作输入端口。 当 VDDR3/VDDR2/VDDR1/VDDR0 位对应的引脚用作外设功能 I/O 引脚时， VDDR3/VDDR2/VDDR1/VDDR0 位的设置被忽略。
	1	GPIO 端口用作输出端口。 当 VDDR3/VDDR2/VDDR1/VDDR0 位对应的引脚用作外设功能 I/O 引脚时， VDDR3/VDDR2/VDDR1/VDDR0 位的设置被忽略。

7.11 端口输入数据寄存器 (VBDIR)

VBDIR 寄存器用于指示引脚的输入数据。

位	7	6	5	4	3	2	1	0
字段	保留				VDIR3	VDIR2	VDIR1	VDIR0
属性	-				R	R	R	R
初始值	-				x	x	x	x

本寄存器所用的接口电路为 TYPE 4。

[bit7:4]保留：保留位

这些位读作 “0b0000”。

写入这些位时，写入“0b0000”。

[bit3] VDIR3: P46/X0A 引脚端口输入数据位

[bit2] VDIR2: P47/X1A 引脚端口输入数据位

[bit1] VDIR1: P49/VWAKEUP 引脚端口输入数据位

[bit0] VDIR0: P48/VREGCTL 引脚端口输入数据位

位		描述
读取	0	不管引脚功能如何设置（VBPFR[3:0]、VBDDR 和 VBDOR 寄存器），本位指示引脚是处于“L”电平输入状态或 “L”电平输出状态。当 P46 和 P47 引脚用作专用功能引脚时（根据 VBPFR 寄存器的 SPSR1 和 SPSR0(bit[5:4])的设置），因输入被封锁本位读取值总为“0”。
	1	不管引脚功能如何设置（VBPFR[3:0]、VBDDR 和 VBDOR 寄存器），本位指示引脚是处于“H”电平输入状态或 “H”电平输出状态。
写入		本位写入操作无效。

7.12 端口输出数据寄存器 (VBDOR)

VBDOR 寄存器用于设置引脚的数据输出。

在 TYPE 2-M0+产品中，P46/X0A 引脚和 P47/X1A 引脚的 GPIO 功能只是输入，因此 VBPCR[3:2]的设置无效。

位	7	6	5	4	3	2	1	0
字段	保留				VDOR3	VDOR2	VDOR1	VDOR0
属性	-				R/W	R/W	R/W	R/W
初始值	-				1	1	1	1

本寄存器所用的接口电路为 TYPE 3。

[bit7:4]保留：保留位

这些位读作 “0b0000”。

写入这些位时，写入“0b0000”。

[bit3] VDOR3: P46/X0A 引脚端口输出数据位

[bit2] VDOR2: P47/X1A 引脚端口输出数据位

[bit1] VDOR1: P49/VWAKEUP 引脚端口输出数据位

[bit0] VDOR0: P48/VREGCTL 引脚端口输出数据位

位		描述
读取		进行读取访问时，读取本位的值。(初始值= 1)
写入	0	输出“L”电平至 GPIO 端口 当此引脚用作输入引脚或外设功能 I/O 引脚时，本位设置被忽略。
	1	输出“H”电平至 GPIO 端口 当此引脚用作输入引脚或外设功能 I/O 引脚时，本位设置被忽略。

7.13 端口伪开漏设置寄存器 (VBPZR)

VBPZR 寄存器用于设置引脚的端口伪开漏。

位	7	6	5	4	3	2	1	0
字段	保留						VPZR1	VPZR0
属性	-						R/W	R/W
初始值	-						1	1

本寄存器所用的接口电路为 TYPE 3。

[bit7:2]保留：保留位

这些位读作“0b000000”。

写入这些位时，写入 “0b000000” 。

[bit1] VPZR1: P49/VWAKEUP 引脚伪开漏设置位

[bit0] VPZR0: P48/VREGCTL 引脚伪开漏设置位

位		描述
读取		进行读取访问时，读取本位的值。(初始值= 1)
写入	0	当 GPIO 端口或外设宏输出数字 “H”电平时，引脚变为“H”电平。
	1	当 GPIO 端口或外设宏输出数字 “H”电平时，引脚变为“Hi-Z”。不管 PCR 寄存器如何设置，上拉电阻器都断开。

8. 使用注意事项

使用备用电源时应注意以下几点：

- 对原电池进行充电或者蓄电池充电过度可能会造成电池泄漏或着火。在确定电池周围电路的配置之前，应检查所用电池的性能。
- 如果板载调节器没有待机引脚，则不能使用睡眠控制功能。如需使用睡眠功能，应选择有待机引脚的产品。

第 7-1 章：中断概述

中断通过 **NVIC** (嵌套向量中断控制器) 聚焦于 **Cortex-M0+** 内核的系统中断控制。

1. 中断参考章节说明

代码：9BFINTTOP_FM0-C03.0

1. 中断参考章节说明

中断配置视产品类型而定。 具体参见以下相关产品对应章节。

Table 1-1 中断对应表

产品类型	参考章节
TYPE 1-M0+	"中断 (TYPE 1)"章 "中断 (TYPE 1)(A)"章 "中断 (TYPE 1)(B)"章
TYPE 2-M0+	"中断 (TYPE 2)"章 "中断 (TYPE 2)(A)"章 "中断 (TYPE 2)(B)"章
TYPE 3-M0+	"中断 (TYPE 3)"章

第 7-2 章：中断 (TYPE 1)

本章说明中断控制器（**TYPE 1**）和外设中断请求（**TYPE 1**）。

1. 概述
2. 配置

代码：9AFIRQC_A-C01.0

1. 概述

中断控制器（TYPE 1）确定中断请求的优先级，并将中断请求发送至 CPU。Cortex-M0+ CPU 内核内部配有嵌套向量中断控制器 (NVIC)。来自若干外设的中断信号被集合并输入至单个中断因素向量。可使用中断请求批量读取寄存器检查已发生的中断请求。此外，某些中断因素情况下，中断请求可转换为 DMA 请求信号。

嵌套向量中断控制器 (NVIC) 的功能特性

- 32 个可屏蔽外设中断通道（不包括 Cortex-M0+ 的 16 个异常中断）
- 4 个可编程中断优先级（使用 2 位优先化中断）
- 促进低时延异常及中断的处理
- 执行系统控制寄存器
- 支持不可屏蔽中断 (NMI) 输入

NVIC 和处理器内核接口紧密耦合，提供低时延中断处理及迟到中断有效处理的使能机制。NVIC 保持嵌套中断信息，使能中断的末尾连锁。

NVIC 管理包括内核异常在内的所有中断。有关异常及 NVIC，详见 ARM 出版的《Cortex-M0+ 技术参考手册》中“第 5 章：异常”及“第 8 章：嵌套向量中断控制器”。

注意事项：

- *《Cortex-M0+ 技术参考手册》中，所有异常类型：IRQ 均定义为外部中断输入。本《手册》中，异常类型：IRQ 定义为外设中断。外设中断包括来自外部引脚的“外部中断和 NMI 控制单元”中断，以及来自 LSI 内的外设的中断。*

中断因素集合功能

将各外设的中断请求信号集合为 32 个源并输入 NVIC。此外，来自外部 NMIX 引脚的中断请求信号通过硬件看门狗中断信号进行逻辑 “或”并输入 NVIC 。

外设中断请求批量读取功能

中断请求批量读取寄存器可将所有外设中断请求信号集合为单一中断请求信号迅速被读出。读**此寄存器能**检查**出已**产**生的中断**请**求**。但此功能不能清除中断请求标志。使用各外设功能的寄存器清除中断请求标志。

DMA 传输请求输出选择功能

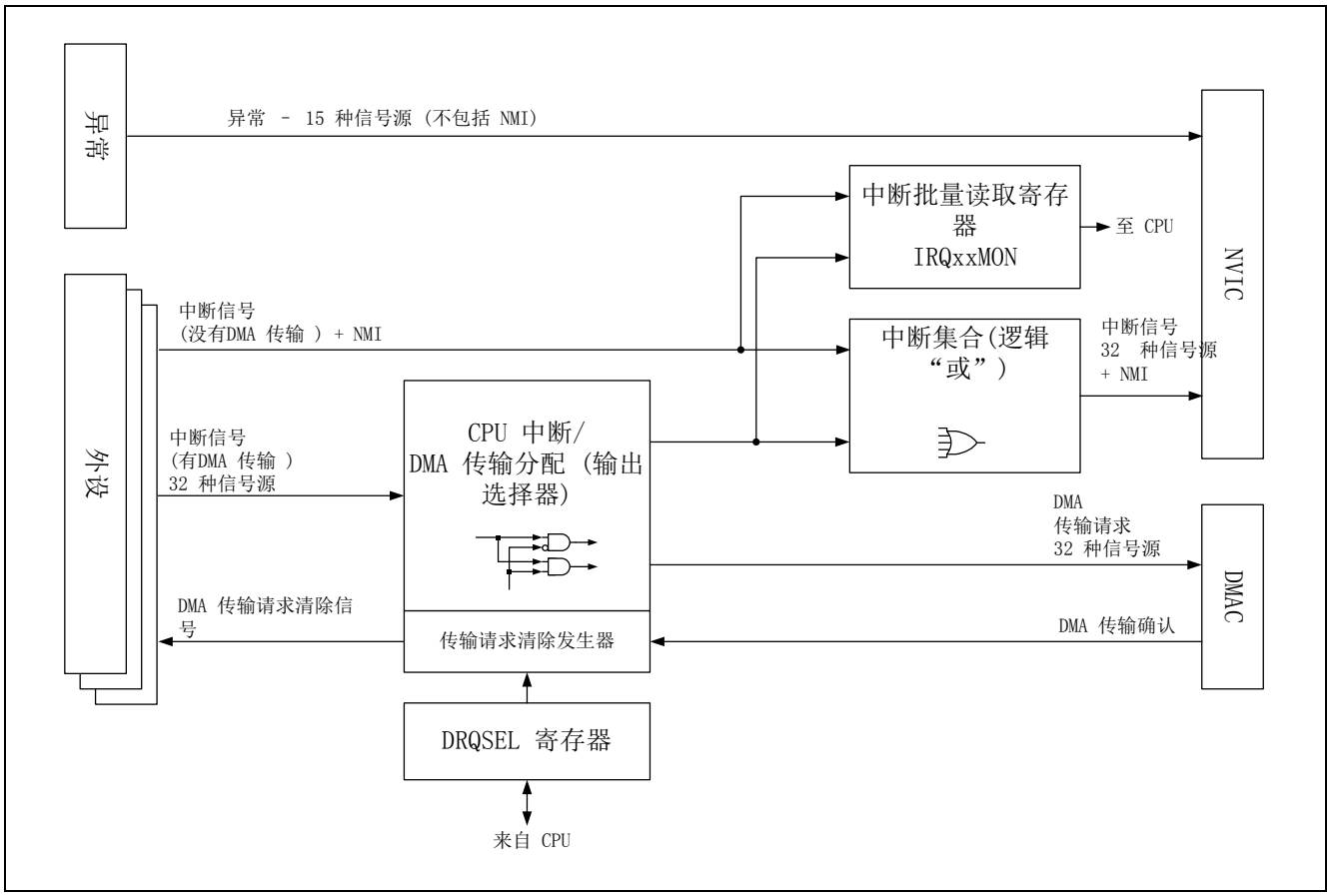
可通过某些外设功能的中断请求激活 DMA 传输。通过 DRQSEL 寄存器，选择将各外设的中断请求信号作为中断请求信号输出至 CPU，或作为传输请求信号输出至 DMAC。有关 DMA 传输请求信号，参见“DMAC”一章。

2. 配置

本节说明中断控制器（TYPE 1）和 DMA 传输请求之间的关系配置。

中断控制器（TYPE-1）和 DMA 传输请求框图

Figure 2-1 中断控制器（TYPE -1）和 DMA 传输请求框图



■ 中断因素集合模块

将各外设的中断请求信号集合（逻辑“或”）为 32 个因素并输出至 NVIC。

■ 外设中断请求批量读取寄存器模块

外设的中断请求信号集合为单一中断请求信号时，此寄存器可检验各外设信号的哪条中断请求生成该中断。

■ CPU 中断请求/DMA 传输请求分配模块

这个输出选择器，通过 DRQSEL 寄存器设置，选择将外设的输出中断请求信号作为中断请求信号输出至 CPU，还是作为传输请求信号输出至 DMAC。

■ 中断因素向量重定位功能

通过 IRQCMODE 寄存器设置可选择 Table 2 所示的两类中断因素向量。有关 IRQCMODE 位,参见“中断 (B)”。各设置的详细内容，参见 Table 1-1 所列各章。

此外，通过 RCINTSEL0 和 RCINTSEL1 寄存器选择任意中断因素。有关 RCINTSEL0 和 RCINTSEL1 寄存器，详见"中断 (B)"。

Table 1-1 中断章节对应表

IRQCMODE 设置	参考章节
IRQCMODE=0 不选择重定位	"中断 (A)"章节
IRQCMODE=1 选择重定位	"中断 (B)"章节

Table 1-2 异常及中断因素向量表

向量号	IRQ 号	异常和中断向量因素	
		IRQCMODE=0	IRQCMODE=1
0	-	栈指针初始值	
1	-	复位	
2	-	不可屏蔽中断 (NMI)/硬件看门狗计时器	
3	-	硬故障	
4	-	保留	
5	-	保留	
6	-	保留	
7 至 10	-	保留	
11	-	SVCall（监视器调用）	
12	-	保留	
13	-	保留	
14	-	PendSV	
15	-	SysTick	
16	0	时钟监视器的异常频率检测 (FCS)	
17	1	软件看门狗计时器	
18	2	低电压检测器 (LVD)	
19	3	MFT 单元 0、单元 1、单元 2 波形发生器 / DTIF（电机紧急停止）	通过 RCINTSEL0 寄存器选择中断因素
20	4	外部引脚中断 ch.0 ~ ch.7	通过 RCINTSEL0 寄存器选择中断因素
21	5	外部引脚中断 ch.8 ~ ch.31	通过 RCINTSEL0 寄存器选择中断因素
22	6	双计时器/四位计数器 (QPRC) ch.0、ch.1、ch.2	通过 RCINTSEL0 寄存器选择中断因素
23	7	MFS ch.0 接收中断/MFS ch.8 接收中断	通过 RCINTSEL1 寄存器选择中断因素
24	8	MFS ch.0 传输中断和状态中断/MFS ch.8 传输中断和状态中断	通过 RCINTSEL1 寄存器选择中断因素
25	9	MFS ch.1 接收中断/MFS ch.9 接收中断	通过 RCINTSEL1 寄存器选择中断因素
26	10	MFS ch.1 传输中断和状态中断/MFS ch.9 传输中断和状态中断	通过 RCINTSEL1 寄存器选择中断因素

向量号	IRQ 号	异常和中断向量因素	
		IRQCMODE=0	IRQCMODE=1
27	11	MFS ch.2 接收中断/MFS ch.10 接收中断	MFT 单元 0 波形发生器/DTIF（电机紧急停止）/MFS ch.8 接收中断、传输中断和状态中断
28	12	MFS ch.2 传输中断和状态中断/MFS ch.10 传输中断和状态中断	外部引脚中断 ch.0 ~ ch.7
29	13	MFS ch.3 接收中断/MFS ch.11 接收中断	外部引脚中断 ch.8 ~ ch.31
30	14	MFS ch.3 传输中断和状态中断/MFS ch.11 传输中断和状态中断	双计时器/四位计时器 (QPRC) ch.0
31	15	MFS ch.4 接收中断/MFS ch.12 接收中断	MFS ch.0 接收中断、传输中断和状态中断
32	16	MFS ch.4 传输中断和状态中断/MFS ch.12 传输中断和状态中断	MFS ch.1 接收中断、传输中断和状态中断
33	17	MFS ch.5 接收中断/MFS ch.13 接收中断	MFS ch.2 接收中断、传输中断和状态中断
34	18	MFS ch.5 传输中断和状态中断/MFS ch.13 传输中断和状态中断	MFS ch.3 接收中断、传输中断和状态中断
35	19	MFS ch.6 接收中断/MFS ch.14 接收中断/ DMA 控制器 (DMAC) ch.0	MFS ch.4 接收中断
36	20	MFS ch.6 传输中断和状态中断/MFS ch.14 传输中断和状态中断/ DMA 控制器 (DMAC) ch.1	MFS ch.4 传输中断和状态中断
37	21	MFS ch.7 接收中断/MFS ch.15 接收中断 / DMA 控制器 (DMAC) ch.2	MFS ch.5 接收中断
38	22	MFS ch.7 传输中断和状态中断/MFS ch.15 传输中断和状态中断 / DMA 控制器 (DMAC) ch.3	MFS ch.5 传输中断和状态中断
39	23	PPG ch.0/2/4/8/10/12/16/18/20	
40	24	外部主 OSC / 外部副 OSC / 主 PLL / 计时计数器/实时计数器	
41	25	A/D 转换器单元 0	A/D 转换器单元 0/MFS ch.9 接收中断、传输中断和状态中断
42	26	A/D 转换器单元 1	A/D 转换器单元 1/MFS ch.10 接收中断、传输中断和状态中断
43	27	A/D 转换器单元 2/LCD 控制器	A/D 转换器单元 2/LCD 控制器/MFS ch.11 接收中断、传输中断和状态中断
44	28	MFT 单元 0、单元 1、单元 2 自由运行计时器	MFT 单元 0 自由运行计时器、输入捕捉、输出比较
45	29	MFT 单元 0、单元 1、单元 2 输入捕捉	MFT 单元 1 自由运行计时器、输入捕捉、输出比较
46	30	MFT 单元 0、单元 1、单元 2 输出比较	MFT 单元 2 自由运行计时器、输入捕捉、输出比较/DMA 控制器 (DMAC) ch.0 ~ ch.7
47	31	基本计时器 ch.0 ~ ch.7 / 闪存 RDY 中断 / 闪存 HANG 中断	

第 7-3 章：中断 (TYPE1-A)

本章说明 **IRQCMODE=0** 时的异常、中断因素向量及寄存器（**TYPE 1**）。

1. 异常及中断因素向量
2. 寄存器
3. 使用注意事项

代码：9AFIRQC_A-C01.0

1. 异常及中断因素向量

本节列出输入至 NVIC 的异常及中断向量表。

Table 1-1 异常及中断因素向量

向量号	IRQ 号	异常及中断因素	向量偏移
0	-	栈指针初始值	0x00
1	-	复位	0x04
2	-	不可屏蔽中断 (NMI)/硬件看门狗计时器	0x08
3	-	硬件故障	0x0C
4	-	保留	0x10
5	-	保留	0x14
6	-	保留	0x18
7-10	-	保留	0x1C - 0x2B
11	-	SVCall（监视器调用）	0x2C
12	-	保留	0x30
13	-	保留	0x34
14	-	PendSV	0x38
15	-	SysTick	0x3C
16	0	时钟监视器的异常频率检测 (FCS)	0x40
17	1	软件看门狗计时器	0x44
18	2	低压检测器 (LVD)	0x48
19	3	MFT 单元 0、单元 1、单元 2 波形发生器 / DTIF（电机紧急停止）	0x4C
20	4	外部引脚中断请求 ch.0 ~ ch.7	0x50
21	5	外部引脚中断请求 ch.8 ~ ch.31	0x54
22	6	双计时器/四位计数器 (QPRC) ch.0、ch.1、ch.2	0x58
23	7	MFS 接收中断请求 ch.0/MFS 接收中断请求 ch.8	0x5C
24	8	MFS 传输中断请求和状态中断请求 ch.0/MFS 传输中断请求和状态中断请求 ch.8	0x60
25	9	MFS 接收中断请求 ch.1/MFS 接收中断请求 ch.9	0x64
26	10	MFS 传输中断请求和状态中断请求 ch.1/MFS 传输中断请求和状态中断请求 ch.9	0x68
27	11	MFS 接收中断请求 ch.2/MFS 接收中断请求 ch.10	0x6C
28	12	MFS 传输中断请求和状态中断请求 ch.2/MFS 传输中断请求和状态中断请求 ch.10	0x70
29	13	MFS 接收中断请求 ch.3/MFS 接收中断请求 ch.11	0x74
30	14	MFS 传输中断请求和状态中断请求 ch.3/MFS 传输中断请求和状态中断请求 ch.11	0x78
31	15	MFS 接收中断请求 ch.4/MFS 接收中断请求 ch.12	0x7C
32	16	MFS 传输中断请求和状态中断请求 ch.4/MFS 传输中断请求和状态中断请求 ch.12	0x80
33	17	MFS 接收中断请求 ch.5/MFS 接收中断请求 ch.13	0x84
34	18	MFS 传输中断请求和状态中断请求 ch.5/MFS 传输中断请求和状态中断请求 ch.13	0x88
35	19	MFS 接收中断请求 ch.6/MFS 接收中断请求 ch.14/DMAC ch.0	0x8C
36	20	MFS 传输中断请求和状态中断请求 ch.6/MFS 传输中断请求和状态中断请求 ch.14/ DMAC ch.1	0x90
37	21	MFS 接收中断请求 ch.7/MFS 接收中断请求 ch.15/DMAC ch.2	0x94

向量号	IRQ 号	异常及中断因素	向量偏移
38	22	MFS 传输中断请求和状态中断请求 ch.7/MFS 传输中断请求和状态中断请求 ch.15/ DMAC ch.3	0x98
39	23	PPG ch.0/2/4/8/10/12/16/18/20	0x9C
40	24	外部主 OSC / 外部副 OSC / 主 PLL / 计时计数器/实时计数器	0xA0
41	25	A/D 转换器单元 0	0xA4
42	26	A/D 转换器单元 1	0xA8
43	27	A/D 转换器单元 2/LCD 控制器	0xAC
44	28	MFT 单元 0、单元 1、单元 2 自由运行计时器	0xB0
45	29	MFT 单元 0、单元 1、单元 2 输入捕捉	0xB4
46	30	MFT 单元 0、单元 1、单元 2 输出比较	0xB8
47	31	基本计时器 ch.0 ~ ch.7 / 闪存 RDY 中断 / 闪存 HANG 中断	0xBC

可使用 NVIC 内置的系统处理程序优先级寄存器（地址 0xE000ED18、0xE000ED1C、0xE000ED20）配置第 4 至第 15 的优先级。可使用 NVIC 内置的 IRQ 优先级寄存器（地址 0xE000E400 至 0xE000E42C）配置第 16 及之后向量的外设中断优先级。

使用批量读取寄存器检查第 2、第 16 至第 47 号向量的中断源。有关其他异常及中断，详见《Cortex-M0+ 技术参考手册》中“第 8 章：嵌套向量中断控制器”。

此外，第 2、第 16 至第 47 号向量时，批量读取的中断源可能是多中断因素在各外设宏内被逻辑“或”的信号。详见各外设资源中断的说明。

2. 寄存器

本节说明 DMA 传输请求选择寄存器和中断请求批量读取寄存器。

DMA 传输请求选择寄存器和中断请求批量读取寄存器表

Table 2-1 DMA 传输请求选择寄存器和中断请求批量读取寄存器表

缩写	寄存器名称	参考章节
DRQSEL	DMA 请求选择寄存器	2.1
EXC02MON	EXC02 批量读取寄存器	2.2
IRQ00MON	IRQ00 批量读取寄存器	2.3
IRQ01MON	IRQ01 批量读取寄存器	2.4
IRQ02MON	IRQ02 批量读取寄存器	2.5
IRQ03MON	IRQ03 批量读取寄存器	2.6
IRQ04MON	IRQ04 批量读取寄存器	2.7
IRQ05MON	IRQ05 批量读取寄存器	2.8
IRQ06MON	IRQ06 批量读取寄存器	2.9
IRQ07MON	IRQ07 批量读取寄存器	2.10
IRQ08MON	IRQ08 批量读取寄存器	2.12
IRQ09MON	IRQ09 批量读取寄存器	2.10
IRQ10MON	IRQ10 批量读取寄存器	2.12
IRQ11MON	IRQ11 批量读取寄存器	2.10
IRQ12MON	IRQ12 批量读取寄存器	2.12
IRQ13MON	IRQ13 批量读取寄存器	2.10
IRQ14MON	IRQ14 批量读取寄存器	2.12
IRQ15MON	IRQ15 批量读取寄存器	2.10
IRQ16MON	IRQ16 批量读取寄存器	2.12
IRQ17MON	IRQ17 批量读取寄存器	2.10
IRQ18MON	IRQ18 批量读取寄存器	2.12
IRQ19MON	IRQ19 批量读取寄存器	2.11
IRQ20MON	IRQ20 批量读取寄存器	2.13
IRQ21MON	IRQ21 批量读取寄存器	2.11
IRQ22MON	IRQ22 批量读取寄存器	2.13
IRQ23MON	IRQ23 批量读取寄存器	2.14
IRQ24MON	IRQ24 批量读取寄存器	2.15
IRQ25MON	IRQ25 批量读取寄存器	2.16
IRQ26MON	IRQ26 批量读取寄存器	2.16
IRQ27MON	IRQ27 批量读取寄存器	2.17
IRQ28MON	IRQ28 批量读取寄存器	2.18
IRQ29MON	IRQ29 批量读取寄存器	2.19
IRQ30MON	IRQ30 批量读取寄存器	2.20
IRQ31MON	IRQ31 批量读取寄存器	2.21

有关 NVIC 内的寄存器，详见《Cortex-M0+ 技术参考手册》中“第 5 章：嵌套向量中断控制器。”

2.1 DMA 请求选择寄存器(DRQSEL)

DMA 请求选择寄存器 (DRQSEL) 选择中断信号是不论是启动 DMA 传输输出作为到 CPU 的中断请求，还是作为输出到 DMAC 的中断请求。

寄存器配置

位	31	5 4 3 2 1 0
字段	DRQSEL[31:5]	保留
属性	R/W	R
初始值	00000000000000000000000000000000	00000

寄存器功能

[bit31:5] DRQSEL:

位号	位	描述
31	0	外部中断 ch.3 的中断信号作为中断请求输出至 CPU。
	1	外部中断 ch.3 的中断信号作为传输请求输出至 DMAC。
30	0	外部中断 ch.2 的中断信号作为中断请求输出至 CPU。
	1	外部中断 ch.2 的中断信号作为传输请求输出至 DMAC。
29	0	外部中断 ch.1 的中断信号作为中断请求输出至 CPU。
	1	外部中断 ch.1 的中断信号作为传输请求输出至 DMAC。
28	0	外部中断 ch.0 的中断信号作为中断请求输出至 CPU。
	1	外部中断 ch.0 的中断信号作为传输请求输出至 DMAC。
27	0	MFS ch.7 的传输中断信号作为中断请求输出至 CPU。
	1	MFS ch.7 的传输中断信号作为传输请求输出至 DMAC。
26	0	MFS ch.7 的接收中断信号作为中断请求输出至 CPU。
	1	MFS ch.7 的接收中断信号作为传输请求输出至 DMAC。
25	0	MFS ch.6 的传输中断信号作为中断请求输出至 CPU。
	1	MFS ch.6 的传输中断信号作为传输请求输出至 DMAC。
24	0	MFS ch.6 的接收中断信号作为中断请求输出至 CPU。
	1	MFS ch.6 的接收中断信号作为传输请求输出至 DMAC。
23	0	MFS ch.5 的传输中断信号作为中断请求输出至 CPU。
	1	MFS ch.5 的传输中断信号作为传输请求输出至 DMAC。
22	0	MFS ch.5 的接收中断信号作为中断请求输出至 CPU。
	1	MFS ch.5 的接收中断信号作为传输请求输出至 DMAC。
21	0	MFS ch.4 的传输中断信号作为中断请求输出至 CPU。
	1	MFS ch.4 的传输中断信号作为传输请求输出至 DMAC。
20	0	MFS ch.4 的接收中断信号作为中断请求输出至 CPU。
	1	MFS ch.4 的接收中断信号作为传输请求输出至 DMAC。
19	0	MFS ch.3 的传输中断信号作为中断请求输出至 CPU。
	1	MFS ch.3 的传输中断信号作为传输请求输出至 DMAC。
18	0	MFS ch.3 的接收中断信号作为中断请求输出至 CPU。
	1	MFS ch.3 的接收中断信号作为传输请求输出至 DMAC。
17	0	MFS ch.2 的传输中断信号作为中断请求输出至 CPU。
	1	MFS ch.2 的传输中断信号作为传输请求输出至 DMAC。

位号	位	描述
16	0	MFS ch.2 的接收中断信号作为中断请求输出至 CPU。
	1	MFS ch.2 的接收中断信号作为传输请求输出至 DMAC。
15	0	MFS ch.1 的传输中断信号作为中断请求输出至 CPU。
	1	MFS ch.1 的传输中断信号作为传输请求输出至 DMAC。
14	0	MFS ch.1 的接收中断信号作为中断请求输出至 CPU。
	1	MFS ch.1 的接收中断信号作为传输请求输出至 DMAC。
13	0	MFS ch.0 的传输中断信号作为中断请求输出至 CPU。
	1	MFS ch.0 的传输中断信号作为传输请求输出至 DMAC。
12	0	MFS ch.0 的接收中断信号作为中断请求输出至 CPU。
	1	MFS ch.0 的接收中断信号作为传输请求输出至 DMAC。
11	0	基本计时器 ch.6 的 IRQ0 中断信号作为中断请求输出至 CPU 。
	1	基本计时器 ch.6 的 IRQ0 中断信号作为传输请求输出至 DMAC（包括扩展）。
10	0	禁止设置。基本计时器 ch.4 的 IRQ0 中断信号作为中断请求输出至 CPU 。
	1	基本计时器 ch.4 的 IRQ0 中断信号作为传输请求输出至 DMAC（包括扩展）。
9	0	基本计时器 ch.2 的 IRQ0 中断信号作为中断请求输出至 CPU。
	1	基本计时器 ch.2 的 IRQ0 中断信号作为传输请求输出至 DMAC。
8	0	基本计时器 ch.0 的 IRQ0 中断信号作为中断请求输出至 CPU 。
	1	基本计时器 ch.0 的 IRQ0 中断信号作为传输请求输出至 DMAC。
7	0	A/D 转换器单元 2 的扫描转换中断信号作为中断请求输出至 CPU。
	1	A/D 转换器单元 2 的扫描转换中断信号作为中断请求输出至 DMAC。
6	0	A/D 转换器单元 1 的扫描转换中断信号作为中断请求输出至 CPU。
	1	A/D 转换器单元 1 的扫描转换中断信号作为中断请求输出至 DMAC。
5	0	A/D 转换器单元 0 的扫描转换中断信号作为中断请求输出至 CPU。
	1	A/D 转换器单元 0 的扫描转换中断信号作为中断请求输出至 DMAC。

MFS: 多功能串口

[bit4:0] 保留：保留位

保留位读为 "0"。

注意事项：

- *若选择中断信号为传输请求发送至 DMAC，无论是否发生中断，相关中断请求批量读取寄存器（IRQxxMON, xx = 00 ~ 31）的读取位值为 “0”。*
- *改变 DRQSEL 设置时，应在更改前清除相关外围设备的中断请求信号。*
- *如果不是 DRQSEL 设置所指定的中断信号，则 DMA 传输不能通过硬件启动。*

2.2 EXC02 批量读取寄存器 (EXC02MON)

EXC02 批量读取寄存器 (EXC02MON) 可批量读取分配给中断因素向量 2 的中断请求。

EXC02MON 指示硬件看门狗计时器和 NMIX 外部引脚的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留														HWINT	NMI
属性	R														R	R
初始值	00000000000000														0	0

寄存器功能

[bit31:2] 保留：保留位

保留位读为 "0"。

[bit1] HWINT:

位	描述
0	没有硬件看门狗计时器中断请求
1	有硬件看门狗计时器中断请求

[bit0] NMI:

位	描述
0	没有 NMIX 外部引脚中断请求
1	有 NMIX 外部引脚中断请求

2.3 IRQ00 批量读取寄存器(IRQ00MON)

IRQ00 批量读取寄存器 (IRQ00MON) 可批量读取分配给中断因素向量 16 的中断请求。

IRQ00MON 指示通过 CSV 进行的异常频率检测的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留															FCSINT
属性	R															R
初始值	000000000000000															0

寄存器功能

[bit31:1] 保留：保留位

保留位读为 “0”。

[bit0] FCSINT:

位	描述
0	没有通过 CSV 中断请求进行的异常频率检测
1	有通过 CSV 中断请求进行的异常频率检测

2.4 IRQ01 批量读取寄存器(IRQ01MON)

IRQ01 批量读取寄存器 (IRQ01MON) 可批量读取分配给中断因素向量 17 的中断请求。

IRQ01MON 指示软件看门狗计时器的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留															SWWDTINT
属性	R															R
初始值	000000000000000															0

寄存器功能

[bit31:1] 保留：保留位

保留位读为 “0”。

[bit0] SWWDTINT:

位	描述
0	没有软件看门狗计时器中断请求
1	有软件看门狗计时器中断请求

2.5 IRQ02 批量读取寄存器 (IRQ02MON)

IRQ02 批量读取寄存器 (IRQ02MON) 可批量读取分配给中断因素向量 18 的中断请求。

IRQ02MON 指示低压检测 (LVD) 的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留															LVDINT
属性	R															R
初始值	000000000000000															0

寄存器功能

[bit31:1] 保留：保留位

保留位读为 “0”。

[bit0] LVDINT:

位	描述
0	没有低压检测 (LVD) 中断请求
1	有低压检测 (LVD) 中断请求

2.6 IRQ03 批量读取寄存器 (IRQ03MON)

IRQ03 批量读取寄存器 (IRQ03MON) 可批量读取分配给中断因素向量 19 的中断请求。

IRQ02MON 指示 MFT 单元 0、MFT 单元 1 和 MFT 单元 2 的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留				WAVE2INT				WAVE1INT				WAVE0INT			
属性	R				R				R				R			
初始值	0000				0000				0000				0000			

寄存器功能

[bit31:12] 保留：保留位

保留位读为 “0”。

[bit11:8] WAVE2INT:

位号	位	描述
11	0	MFT 单元 2 内没有 WFG 计时器 54 中断请求
	1	MFT 单元 2 内有 WFG 计时器 54 中断请求
10	0	MFT 单元 2 内没有 WFG 计时器 32 中断请求
	1	MFT 单元 2 内有 WFG 计时器 32 中断请求
9	0	MFT 单元 2 内没有 WFG 计时器 10 中断请求
	1	MFT 单元 2 内有 WFG 计时器 10 中断请求
8	0	MFT 单元 2 内没有 DTIF（电机紧急停止）中断请求
	1	MFT 单元 2 内有 DTIF（电机紧急停止）中断请求

[bit7:4] WAVE1INT:

位号	位	描述
7	0	MFT 单元 1 内没有 WFG 计时器 54 中断请求
	1	MFT 单元 1 内有 WFG 计时器 54 中断请求
6	0	MFT 单元 1 内没有 WFG 计时器 32 中断请求
	1	MFT 单元 1 内有 WFG 计时器 32 中断请求
5	0	MFT 单元 1 内没有 WFG 计时器 10 中断请求
	1	MFT 单元 1 内有 WFG 计时器 10 中断请求
4	0	MFT 单元 1 内没有 DTIF（电机紧急停止）中断请求
	1	MFT 单元 1 内有 DTIF（电机紧急停止）中断请求

[bit3:0] WAVE0INT:

位号	位	描述
3	0	MFT 单元 0 内没有 WFG 计时器 54 中断请求
	1	MFT 单元 0 内有 WFG 计时器 54 中断请求
2	0	MFT 单元 0 内没有 WFG 计时器 32 中断请求
	1	MFT 单元 0 内有 WFG 计时器 32 中断请求
1	0	MFT 单元 0 内没有 WFG 计时器 10 中断请求
	1	MFT 单元 0 内有 WFG 计时器 10 中断请求
0	0	MFT 单元 0 内没有 DTIF（电机紧急停止）中断请求
	1	MFT 单元 0 内有 DTIF（电机紧急停止）中断请求

2.7 IRQ04 批量读取寄存器 (IRQ04MON)

IRQ04 批量读取寄存器 (IRQ04MON) 可批量读取分配给中断因素向量 20 的中断请求。

IRQ04MON 指示外部中断 ch.0 ~ ch.7 的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留								EXTINT							
属性	R								R							
初始值	0x00								0x00							

寄存器功能

[bit31:8] 保留：保留位

保留位读为 “0”。

[bit7:0] EXTINT:

位号	位	描述
7	0	没有外部中断 ch.7 的中断请求
	1	有外部中断 ch.7 的中断请求
6	0	没有外部中断 ch.6 的中断请求
	1	有外部中断 ch.6 的中断请求
5	0	没有外部中断 ch.5 的中断请求
	1	有外部中断 ch.5 的中断请求
4	0	没有外部中断 ch.4 的中断请求
	1	有外部中断 ch.4 的中断请求
3	0	没有外部中断 ch.3 的中断请求
	1	有外部中断 ch.3 的中断请求
2	0	没有外部中断 ch.2 的中断请求
	1	有外部中断 ch.2 的中断请求
1	0	没有外部中断 ch.1 的中断请求
	1	有外部中断 ch.1 的中断请求
0	0	没有外部中断 ch.0 的中断请求
	1	有外部中断 ch.0 的中断请求

如果通过 DRQSEL 寄存器选择 DMA 传输请求，对应的 EXTINT 位为 "0"。

2.8 IRQ05 批量读取寄存器 (IRQ05MON)

IRQ05 批量读取寄存器 (IRQ05MON) 可批量读取分配给中断因素向量 21 的中断请求。

IRQ05MON 指示外部中断 ch.8 ~ ch.31 的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留								EXTINT							
属性	R								R							
初始值	0x00								0x00							

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	EXTINT															
属性	R															
初始值	0x0000															

寄存器功能

[bit31:24] 保留：保留位

保留位读为 “0”。

[bit23:0] EXTINT:

位号	位	描述
23	0	没有外部中断 ch.31 的中断请求
	1	有外部中断 ch.31 的中断请求
22	0	没有外部中断 ch.30 的中断请求
	1	有外部中断 ch.30 的中断请求
21	0	没有外部中断 ch.29 的中断请求
	1	有外部中断 ch.29 的中断请求
20	0	没有外部中断 ch.28 的中断请求
	1	有外部中断 ch.28 的中断请求
19	0	没有外部中断 ch.27 的中断请求
	1	有外部中断 ch.27 的中断请求
18	0	没有外部中断 ch.26 的中断请求
	1	有外部中断 ch.26 的中断请求
17	0	没有外部中断 ch.25 的中断请求
	1	有外部中断 ch.25 的中断请求
16	0	没有外部中断 ch.24 的中断请求
	1	有外部中断 ch.24 的中断请求
15	0	没有外部中断 ch.23 的中断请求
	1	有外部中断 ch.23 的中断请求
14	0	没有外部中断 ch.22 的中断请求
	1	有外部中断 ch.22 的中断请求
13	0	没有外部中断 ch.21 的中断请求
	1	有外部中断 ch.21 的中断请求

位号	位	描述
12	0	没有外部中断 ch.20 的中断请求
	1	有外部中断 ch.20 的中断请求
11	0	没有外部中断 ch.19 的中断请求
	1	有外部中断 ch.19 的中断请求
10	0	没有外部中断 ch.18 的中断请求
	1	有外部中断 ch.18 的中断请求
9	0	没有外部中断 ch.17 的中断请求
	1	有外部中断 ch.17 的中断请求
8	0	没有外部中断 ch.16 的中断请求
	1	有外部中断 ch.16 的中断请求
7	0	没有外部中断 ch.15 的中断请求
	1	有外部中断 ch.15 的中断请求
6	0	没有外部中断 ch.14 的中断请求
	1	有外部中断 ch.14 的中断请求
5	0	没有外部中断 ch.13 的中断请求
	1	有外部中断 ch.13 的中断请求
4	0	没有外部中断 ch.12 的中断请求
	1	有外部中断 ch.12 的中断请求
3	0	没有外部中断 ch.11 的中断请求
	1	有外部中断 ch.11 的中断请求
2	0	没有外部中断 ch.10 的中断请求
	1	有外部中断 ch.10 的中断请求
1	0	没有外部中断 ch.9 的中断请求
	1	有外部中断 ch.9 的中断请求
0	0	没有外部中断 ch.8 的中断请求
	1	有外部中断 ch.8 的中断请求

2.9 IRQ06 批量读取寄存器 (IRQ06MON)

IRQ06 批量读取寄存器 (IRQ06MON) 可批量读取分配给中断因素向量 22 的中断请求。

IRQ06MON 指示 QPRC 和双计时器的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留												QUD2INT			
属性	R												R			
初始值	0x000												0000			

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	QUD2INT		QUD1INT						QUD0INT						TIMINT	
属性	R		R						R						R	
初始值	0x0000		000000						000000						00	

寄存器功能

[bit31:20] 保留：保留位

保留位读为 “0”。

[bit19:14] QUD2INT:

位号	位	描述
19	0	没有 QPRC ch.2 的 PC 匹配和 RC 匹配中断请求
	1	有 QPRC ch.2 的 PC 匹配和 RC 匹配中断请求
18	0	QPRC ch.2 没有检测到范围外 RC 的中断请求
	1	QPRC ch.2 检测到范围外 RC 的中断请求
17	0	没有 QPRC ch.2 的 PC 计数反向中断请求
	1	有 QPRC ch.2 的 PC 计数反向中断请求
16	0	没有 QPRC ch.2 的上溢/下溢/零索引中断请求
	1	有 QPRC ch.2 的上溢/下溢/零索引中断请求
15	0	没有 QPRC ch.2 的 PC 和 RC 匹配中断请求
	1	有 QPRC ch.2 的 PC 和 RC 匹配中断请求
14	0	没有 QPRC ch.2 的 PC 匹配中断请求
	1	有 QPRC ch.2 的 PC 匹配中断请求

[bit13:8] QUD1INT:

位号	位	描述
13	0	没有 QPRC ch.1 的 PC 匹配和 RC 匹配中断请求
	1	有 QPRC ch.1 的 PC 匹配和 RC 匹配中断请求
12	0	QPRC ch.1 没有检测到范围外 RC 的中断请求
	1	QPRC ch.1 检测到范围外 RC 的中断请求
11	0	没有 QPRC ch.1 PC 计数反向中断请求
	1	有 QPRC ch.1 的 PC 计数反向中断请求
10	0	没有 QPRC ch.1 的上溢/下溢/零索引中断请求
	1	有 QPRC ch.1 的上溢/下溢/零索引中断请求
9	0	没有 QPRC ch.1 的 PC 和 RC 匹配中断请求
	1	有 QPRC ch.1 的 PC 和 RC 匹配中断请求
8	0	没有 QPRC ch.1 的 PC 匹配中断请求
	1	有 QPRC ch.1 的 PC 匹配中断请求

[bit7:2] QUD0INT:

位号	位	描述
7	0	没有 QPRC ch.0 的 PC 匹配和 RC 匹配中断请求
	1	有 QPRC ch.0 的 PC 匹配和 RC 匹配中断请求
6	0	QPRC ch.0 没有检测到范围外 RC 的中断请求
	1	QPRC ch.0 检测到范围外 RC 的中断请求
5	0	没有 QPRC ch.0 的 PC 计数反向中断请求
	1	有 QPRC ch.0 的 PC 计数反向中断请求
4	0	没有 QPRC ch.0 的上溢/下溢/零索引中断请求
	1	有 QPRC ch.0 的上溢/下溢/零索引中断请求
3	0	没有 QPRC ch.0 的 PC 和 RC 匹配中断请求
	1	有 QPRC ch.0 的 PC 和 RC 匹配中断请求
2	0	没有 QPRC ch.0 的 PC 匹配中断请求
	1	有 QPRC ch.0 的 PC 匹配中断请求

[bit1:0] TIMINT:

位号	位	描述
1	0	没有双计时器 TIMINT2 中断请求
	1	有双计时器 TIMINT2 中断请求
0	0	没有双计时器 TIMINT1 中断请求
	1	有双计时器 TIMINT1 中断请求

2.10 IRQ07/09/11/13/15/17 批量读取寄存器(IRQxxMON)

IRQ07 批量读取寄存器 (IRQ07MON) 可批量读取分配给中断因素向量 23 的中断请求。
IRQ09 批量读取寄存器(IRQ09MON)可批量读取分配给中断因素向量 25 的中断请求。
IRQ11 批量读取寄存器 (IRQ11MON) 可批量读取分配给中断因素向量 27 的中断请求。
IRQ13 批量读取寄存器 (IRQ13MON) 可批量读取分配给中断因素向量 29 的中断请求。
IRQ15 批量读取寄存器 (IRQ15MON) 可批量读取分配给中断因素向量 31 的中断请求。
IRQ17 批量读取寄存器 (IRQ17MON) 批量读取分配给中断因素向量 33 的中断请求。

IRQ07MON 指示 MFS ch.0 / ch.8 的接收中断请求状态。
IRQ09MON 指示 MFS ch.1 / ch.9 的接收中断请求状态。
IRQ11MON 指示 MFS ch.2 / ch.10 的接收中断请求状态。
IRQ13MON 指示 MFS ch.3 / ch.11 的接收中断请求状态。
IRQ15MON 指示 MFS ch.4 / ch.12 的接收中断请求状态。
IRQ17MON 指示 MFS ch.5 / ch.13 的接收中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留														MFSINT	
属性	R														R	
初始值	00000000000000														00	

寄存器功能

[bit31:2] 保留：保留位
保留位读为 “0”。

[bit1:0] MFSINT:

位号	位	描述
1	0	对应 MFS 通道 (ch.8 ~ ch.13) 没有接收中断请求
	1	对应 MFS 通道 (ch.8 ~ ch.13) 有接收中断请求
0	0	对应 MFS 通道 (ch.0 ~ ch.5) 没有接收中断请求
	1	对应 MFS 通道 (ch.0 ~ ch.5) 有接收中断请求

如果 DRQSEL 寄存器已选择 DMA 传输请求，对应 MFS 通道（ch.0 ~ ch.5）的 MFSINT 位为 "0"。

2.11 IRQ19/21 批量读取寄存器 (IRQxxMON)

IRQ19 批量读取寄存器 (IRQ19MON) 可批量读取分配给中断因素向量 35 的中断请求。
IRQ21 批量读取寄存器 (IRQ21MON) 可批量读取分配给中断因素向量 37 的中断请求。

IRQ19MON 指示 MFS ch.6 / ch.14 的接收中断请求状态及 DMAC ch.0 的中断请求状态。
IRQ21MON 指示 MFS ch.7 / ch.15 的接收中断请求状态及 DMAC ch.2 的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留											DMAINT	保留		MFSINT	
属性	R											R	R		R	
初始值	00000000000											0	00		00	

寄存器功能

[bit31:5] 保留：保留位
保留位读为 “0”。

[bit4] DMAINT:

位	描述
1	没有 DMAC ch.0 / ch.2 的中断请求
	有 DMAC ch.0 / ch.2 的中断请求

[bit3:2] 保留：保留位
保留位读为 “0”。

[bit1:0] MFSINT:

位号	位	描述
1	0	对应 MFS 通道 (ch.14、 ch.15) 没有接收中断请求
	1	对应 MFS 通道 (ch.14、 ch.15) 有接收中断请求
0	0	对应 MFS 通道 (ch.6、 ch.7) 没有接收中断请求
	1	对应 MFS 通道 (ch.6、ch.7) 有接收中断请求

如果 DRQSEL 寄存器已选择 DMA 传输请求，对应的 MFS 通道 (ch.6、ch.7) 的 MFSINT 位为 "0"。

2.12 IRQ08/10/12/14/16/18 批量读取寄存器 (IRQxxMON)

IRQ08 批量读取寄存器 (IRQ08MON) 批量读取分配给中断因素向量 24 的中断请求。
IRQ10 批量读取寄存器 (IRQ10MON) 可批量读取分配给中断因素向量 26 的中断请求。
IRQ12 批量读取寄存器 (IRQ12MON) 可批量读取分配给中断因素向量 28 的中断请求。
IRQ14 批量读取寄存器 (IRQ14MON) 可批量读取分配给中断因素向量 30 的中断请求。
IRQ16 批量读取寄存器 (IRQ16MON) 可批量读取分配给中断因素向量 32 的中断请求。
IRQ18 批量读取寄存器 (IRQ18MON) 可批量读取分配给中断因素向量 34 的中断请求。

IRQ08MON 指示 MFS ch.0 和 ch.8 的传输中断请求和状态中断请求的状态。
IRQ10MON 指示 MFS ch.1 和 ch.9 的传输中断请求和状态中断请求的状态。
IRQ12MON 指示 MFS ch.2 和 ch.10 的传输中断请求和状态中断请求的状态。
IRQ14MON 指示 MFS ch.3 和 ch.11 的传输中断请求和状态中断请求的状态。
IRQ16MON 指示 MFS ch.4 和 ch.12 的传输中断请求和状态中断请求的状态。
IRQ18MON 指示 MFS ch.5 和 ch.13 的传输中断请求和状态中断请求的状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留												MFSINT			
属性	R												R			
初始值	00000000000000												0000			

寄存器功能

[bit31:4] 保留：保留位
保留位读为 “0”。

[bit3:0] MFSINT:

位号	位	描述
3	0	对应 MFS 通道 (ch.8 ~ ch.13) 没有状态中断请求
	1	对应 MFS 通道 (ch.8 ~ ch.13) 有状态中断请求
2	0	对应 MFS 通道 (ch.8 ~ ch.13) 没有传输中断请求
	1	对应 MFS 通道 (ch.8 ~ ch.13) 有传输中断请求
1	0	对应 MFS 通道 (ch.0 ~ ch.5) 没有状态中断请求
	1	对应 MFS 通道 (ch.0 ~ ch.5) 有状态中断请求
0	0	对应 MFS 通道 (ch.0 ~ ch.5) 没有传输中断请求
	1	对应 MFS 通道 (ch.0 ~ ch.5) 有传输中断请求

如果 DRQSEL 寄存器已选择 DMA 传输请求，对应 MFS 通道（ch.0 ~ ch.5）的 MFSINT 位为 "0"。

2.13 IRQ20/22 批量读取寄存器 (IRQxxMON)

IRQ20 批量读取寄存器 (IRQ20MON) 批量读取分配给中断因素向量 36 的中断请求。
IRQ22 批量读取寄存器 (IRQ22MON) 批量读取分配给中断因素向量 38 的中断请求。

IRQ20MON 指示 MFS ch.6 / ch.14 的传输中断请求的状态、状态中断请求的状态及 DMAC ch.1 的中断请求状态。
IRQ22MON 表明 MFS ch.7 / ch.15 的传输中断请求的状态、状态中断请求的状态以及 DMAC ch.3 的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留											DMAINT	MFSINT			
属性	R											R	R			
初始值	00000000000											0	0000			

寄存器功能

[bit31:5] 保留：保留位
保留位读为 “0”。

[bit4] DMAINT:

位	描述
1	没有 DMAC ch.1/ ch.3 的中断请求
	有 DMAC ch.1 / ch.3 的中断请求

[bit3:0] MFSINT:

位号	位	描述
3	0	对应 MFS 通道 (ch.14、ch.15) 没有状态中断请求
	1	对应 MFS 通道 (ch.14、ch.15) 有状态中断请求
2	0	对应 MFS 通道 (ch.14、ch.15) 没有传输中断请求
	1	对应 MFS 通道 (ch.14、ch.15) 有传输中断请求
1	0	对应 MFS 通道 (ch.6、ch.7) 没有状态中断请求
	1	对应 MFS 通道 (ch.6、ch.7) 有状态中断请求
0	0	对应 MFS 通道 (ch.6、ch.7) 没有传输中断请求
	1	对应 MFS 通道 (ch.6、ch.7) 有传输中断请求

如果 DRQSEL 寄存器已选择 DMA 传输请求，对应的 MFS 通道 (ch.6、ch.7) 的 MFSINT 位为 "0"。

2.14 IRQ23 批量读取寄存器 (IRQ23MON)

IRQ23 批量读取寄存器 (IRQ23MON) 批量读取分配给中断因素向量 39 的中断请求。

IRQ23MON 指示 PPG 的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留							PPGINT								
属性	R							R								
初始值	0000000							000000000								

寄存器功能

[bit31:9] 保留：保留位

保留位读为 “0”。

[bit8:0] PPGINT:

位号	位	描述
8	0	没有 PPG ch.20 的中断请求
	1	有 PPG ch.20 的中断请求
7	0	没有 PPG ch.18 的中断请求
	1	有 PPG ch.18 的中断请求
6	0	没有 PPG ch.16 的中断请求
	1	有 PPG ch.16 的中断请求
5	0	没有 PPG ch.12 的中断请求
	1	有 PPG ch.12 的中断请求
4	0	没有 PPG ch.10 的中断请求
	1	有 PPG ch.10 的中断请求
3	0	没有 PPG ch.8 的中断请求
	1	有 PPG ch.8 的中断请求
2	0	没有 PPG ch.4 的中断请求
	1	有 PPG ch.4 的中断请求
1	0	没有 PPG ch.2 的中断请求
	1	有 PPG ch.2 的中断请求
0	0	没有 PPG ch.0 的中断请求
	1	有 PPG ch.0 的中断请求

2.15 IRQ24 批量读取寄存器 (IRQ24MON)

IRQ24 批量读取寄存器 (IRQ24MON) 可批量读取分配给中断因素向量 40 的中断请求。

IRQ24MON 指示 RTC、计时计数器、主 PLL 振荡、副振荡和主时钟振荡的中断请求状态。

寄存器配置

位	31							8
字段	保留							
属性	R							
初始值	0x000000							

位	7	6	5	4	3	2	1	0
字段	保留		RTCINT	WCINT	保留	MPLLINT	SOSCINT	MOSCINT
属性	R		R	R	R	R	R	R
初始值	00		0	0	0	0	0	0

寄存器功能

[bit31:6] 保留：保留位

保留位读为 “0”。

[bit5] RTCINT:

位	描述
0	没有 RTC 中断请求
1	有 RTC 中断请求

[bit4] WCINT:

位	描述
0	没有计时计数器中断请求
1	有计时计数器中断请求

[bit3] 保留：保留位

保留位读为 “0”。

[bit2] MPLLINT:

位	描述
0	没有主 PLL 振荡的稳定等待完成中断请求
1	有主 PLL 振荡的稳定等待完成中断请求

[bit1] SOSCINT:

位	描述
0	没有副时钟振荡的稳定等待完成中断请求
1	有副时钟振荡的稳定等待完成中断请求

[bit0] MOSCINT:

位	描述
0	没有主时钟振荡的稳定等待完成中断请求
1	有主时钟振荡的稳定等待完成中断请求

2.16 IRQ25/26 批量读取寄存器(IRQxxMON)

IRQ25 批量读取寄存器 (IRQ25MON) 可批量读取分配给中断因素向量 41 的中断请求。
IRQ26 批量读取寄存器 (IRQ26MON) 可批量读取分配给中断因素向量 42 的中断请求。

IRQ25MON 指示 A/D 转换器单元 0 的中断请求状态。
IRQ26MON 指示 A/D 转换器单元 1 的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留											ADCINT				
属性	R											R				
初始值	00000000000											00000				

寄存器功能

[bit31:5] 保留：保留位
保留位读为 “0”。

[bit4:0] ADCINT:

位号	位	描述
4	0	没有对应的 A/D 转换器单元的范围比较结果中断请求
	1	有对应的 A/D 转换器单元的范围比较结果中断请求
3	0	没有对应的 A/D 转换器单元的转换结果比较中断请求
	1	有对应的 A/D 转换器单元的转换结果比较中断请求
2	0	没有对应的 A/D 转换器单元的 FIFO 溢出中断请求
	1	有对应的 A/D 转换器单元的 FIFO 溢出中断请求
1	0	没有对应的 A/D 转换器单元的扫描转换中断请求
	1	有对应的 A/D 转换器单元的扫描转换中断请求
0	0	没有对应的 A/D 转换器单元的优先转换中断请求
	1	有对应的 A/D 转换器单元的优先转换中断请求

如果 DRQSEL 寄存器已选择 DMA 传输请求，对应的 ADCINT 位为 "0"。

2.17 IRQ27 批量读取寄存器(IRQ27MON)

IRQ27 批量读取寄存器 (IRQ27MON) 可批量读取分配给中断因素向量 43 的中断请求。

IRQ27MON 指示 A/D 转换器单元 2 和 LCD 控制器的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留										LCDCINT	ADCINT				
属性	R										R	R				
初始值	0000000000										0	00000				

寄存器功能

[bit31:6] 保留：保留位

保留位读为 “0”。

[bit5] LCDCINT:

位	描述
0	没有 LCD 控制器的中断请求
1	有 LCD 控制器的中断请求

[bit4:0] ADCINT:

位号	位	描述
4	0	没有对应的 A/D 转换器单元 2 的范围比较结果中断请求
	1	有对应的 A/D 转换器单元 2 的范围比较结果中断请求
3	0	没有 A/D 转换器单元 2 的转换结果比较中断请求
	1	有 A/D 转换器单元 2 的转换结果比较中断请求
2	0	没有 A/D 转换器单元 2 的 FIFO 溢出中断请求
	1	有 A/D 转换器单元 2 的 FIFO 溢出中断请求
1	0	没有 A/D 转换器单元 2 的扫描转换中断请求
	1	有 A/D 转换器单元 2 的扫描转换中断请求
0	0	没有 A/D 转换器单元 2 的优先转换中断请求
	1	有 A/D 转换器单元 2 的优先转换中断请求

如果通过 DRQSEL 寄存器选择 DMA 传输请求，对应的 ADCINT 位为 "0"。

2.18 IRQ28 批量读取寄存器 (IRQ28MON)

IRQ28 批量读取寄存器 (IRQ28MON) 批量读取分配给中断因素向量 44 的中断请求。

IRQ28MON 指示 MFT 单元 0、MFT 单元 1 和 MFT 单元 2 的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留														FRT2INT	
属性	R														R	
初始值	00000000000000														00	

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	FRT2INT				FRT1INT						FRT0INT					
属性	R				R						R					
初始值	0000				000000						000000					

寄存器功能

[bit31:18] 保留：保留位

保留位读为 “0”。

[bit17:12] FRT2INT:

位号	位	描述
17	0	MFT 单元 2 内自由运行计时器 ch.2 没有零检测中断请求
	1	MFT 单元 2 内自由运行计时器 ch.2 有零检测中断请求
16	0	MFT 单元 2 内自由运行计时器 ch.1 没有零检测中断请求
	1	MFT 单元 2 内自由运行计时器 ch.1 有零检测中断请求
15	0	MFT 单元 2 内自由运行计时器 ch.0 没有零检测中断请求
	1	MFT 单元 2 内自由运行计时器 ch.0 有零检测中断请求
14	0	MFT 单元 2 内自由运行计时器 ch.2 没有峰值检测中断请求
	1	MFT 单元 2 内自由运行计时器 ch.2 有峰值检测中断请求
13	0	MFT 单元 2 内自由运行计时器 ch.1 没有峰值检测中断请求
	1	MFT 单元 2 内自由运行计时器 ch.1 有峰值检测中断请求
12	0	MFT 单元 2 内自由运行计时器 ch.0 没有峰值检测中断请求
	1	MFT 单元 2 内自由运行计时器 ch.0 有峰值检测中断请求

[bit11:6] FRT1INT:

位号	位	描述
11	0	MFT 单元 1 内自由运行计时器 ch.2 没有零检测中断请求
	1	MFT 单元 1 内自由运行计时器 ch.2 有零检测中断请求
10	0	MFT 单元 1 内自由运行计时器 ch.1 没有零检测中断请求
	1	MFT 单元 1 内自由运行计时器 ch.1 有零检测中断请求
9	0	MFT 单元 1 内自由运行计时器 ch.0 没有零检测中断请求
	1	MFT 单元 1 内自由运行计时器 ch.0 有零检测中断请求
8	0	MFT 单元 1 内自由运行计时器 ch.2 没有峰值检测中断请求
	1	MFT 单元 1 内自由运行计时器 ch.2 有峰值检测中断请求
7	0	MFT 单元 1 内自由运行计时器 ch.1 没有峰值检测中断请求
	1	MFT 单元 1 内自由运行计时器 ch.1 有峰值检测中断请求
6	0	MFT 单元 1 内自由运行计时器 ch.0 没有峰值检测中断请求
	1	MFT 单元 1 内自由运行计时器 ch.0 有峰值检测中断请求

[bit5:0] FRT0INT:

位号	位	描述
5	0	MFT 单元 0 内自由运行计时器 ch.2 没有零检测中断请求
	1	MFT 单元 0 内自由运行计时器 ch.2 有零检测中断请求
4	0	MFT 单元 0 内自由运行计时器 ch.1 没有零检测中断请求
	1	MFT 单元 0 内自由运行计时器 ch.1 有零检测中断请求
3	0	MFT 单元 0 内自由运行计时器 ch.0 没有零检测中断请求
	1	MFT 单元 0 内自由运行计时器 ch.0 有零检测中断请求
2	0	MFT 单元 0 内自由运行计时器 ch.2 没有峰值检测中断请求
	1	MFT 单元 0 内自由运行计时器 ch.2 有峰值检测中断请求
1	0	MFT 单元 0 内自由运行计时器 ch.1 没有峰值检测中断请求
	1	MFT 单元 0 内自由运行计时器 ch.1 有峰值检测中断请求
0	0	MFT 单元 0 内自由运行计时器 ch.0 没有峰值检测中断请求
	1	MFT 单元 0 内自由运行计时器 ch.0 有峰值检测中断请求

2.19 IRQ29 批量读取寄存器 (IRQ29MON)

IRQ29 批量读取寄存器 (IRQ29MON) 可批量读取分配给中断因素向量 45 的中断请求。

IRQ29MON 指示 MFT 单元 0、MFT 单元 1 和 MFT 单元 2 的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留				ICU2INT				ICU1INT				ICU0INT			
属性	R				R				R				R			
初始值	0000				0000				0000				0000			

寄存器功能

[bit31:12] 保留：保留位

保留位读为 “0”。

[bit11:8] ICU2INT:

位号	位	描述
11	0	MFT 单元 2 的输入捕捉 ch.3 没有中断请求
	1	MFT 单元 2 的输入捕捉 ch.3 有中断请求
10	0	MFT 单元 2 的输入捕捉 ch.2 没有中断请求
	1	MFT 单元 2 的输入捕捉 ch.2 有中断请求
9	0	MFT 单元 2 的输入捕捉 ch.1 没有中断请求
	1	MFT 单元 2 的输入捕捉 ch.1 有中断请求
8	0	MFT 单元 2 的输入捕捉 ch.0 没有中断请求
	1	MFT 单元 2 的输入捕捉 ch.0 有中断请求

[bit7:4] ICU1INT:

位号	位	描述
7	0	MFT 单元 1 的输入捕捉 ch.3 没有中断请求
	1	MFT 单元 1 的输入捕捉 ch.3 有中断请求
6	0	MFT 单元 1 的输入捕捉 ch.2 没有中断请求
	1	MFT 单元 1 的输入捕捉 ch.2 有中断请求
5	0	MFT 单元 1 的输入捕捉 ch.1 没有中断请求
	1	MFT 单元 1 的输入捕捉 ch.1 有中断请求
4	0	MFT 单元 1 的输入捕捉 ch.0 没有中断请求
	1	MFT 单元 1 的输入捕捉 ch.0 有中断请求

[bit3:0] ICU0INT:

位号	位	描述
3	0	MFT 单元 0 的输入捕捉 ch.3 没有中断请求
	1	MFT 单元 0 的输入捕捉 ch.3 有中断请求
2	0	MFT 单元 0 的输入捕捉 ch.2 没有中断请求
	1	MFT 单元 0 的输入捕捉 ch.2 有中断请求
1	0	MFT 单元 0 的输入捕捉 ch.1 没有中断请求
	1	MFT 单元 0 的输入捕捉 ch.1 有中断请求
0	0	MFT 单元 0 的输入捕捉 ch.0 没有中断请求
	1	MFT 单元 0 的输入捕捉 ch.0 有中断请求

2.20 IRQ30 批量读取寄存器 (IRQ30MON)

IRQ30 批量读取寄存器 (IRQ30MON) 可批量读取分配给中断因素向量 46 的中断请求。

IRQ30MON 指示 MFT 单元 0、MFT 单元 1 和 MFT 单元 2 的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留														OCU2INT	
属性	R														R	
初始值	00000000000000														00	

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	OCU2INT				OCU1INT						OCU0INT					
属性	R				R						R					
初始值	0000				000000						000000					

寄存器功能

[bit31:18] 保留：保留位

保留位读为 “0”。

[bit17:12] OCU2INT:

位号	位	描述
17	0	MFT 单元 2 内输出比较 ch.5 没有中断请求
	1	MFT 单元 2 内输出比较 ch.5 有中断请求
16	0	MFT 单元 2 内输出比较 ch.4 没有中断请求
	1	MFT 单元 2 内输出比较 ch.4 有中断请求
15	0	MFT 单元 2 内输出比较 ch.3 没有中断请求
	1	MFT 单元 2 内输出比较 ch.3 有中断请求
14	0	MFT 单元 2 内输出比较 ch.2 没有中断请求
	1	MFT 单元 2 内输出比较 ch.2 有中断请求
13	0	MFT 单元 2 内输出比较 ch.1 没有中断请求
	1	MFT 单元 2 内输出比较 ch.1 有中断请求
12	0	MFT 单元 2 内输出比较 ch.0 没有中断请求
	1	MFT 单元 2 内输出比较 ch.0 有中断请求

[bit11:6] OCU1INT:

位号	位	描述
11	0	MFT 单元 1 内输出比较 ch.5 没有中断请求
	1	MFT 单元 1 内输出比较 ch.5 有中断请求
10	0	MFT 单元 1 内输出比较 ch.4 没有中断请求
	1	MFT 单元 1 内输出比较 ch.4 有中断请求
9	0	MFT 单元 1 内输出比较 ch.3 没有中断请求
	1	MFT 单元 1 内输出比较 ch.3 有中断请求
8	0	MFT 单元 1 内输出比较 ch.2 没有中断请求
	1	MFT 单元 1 内输出比较 ch.2 有中断请求
7	0	MFT 单元 1 内输出比较 ch.1 没有中断请求
	1	MFT 单元 1 内输出比较 ch.1 有中断请求
6	0	MFT 单元 1 内输出比较 ch.0 没有中断请求
	1	MFT 单元 1 内输出比较 ch.0 有中断请求

[bit5:0] OCU0INT:

位号	位	描述
5	0	MFT 单元 0 内输出比较 ch.5 没有中断请求
	1	MFT 单元 0 内输出比较 ch.5 有中断请求
4	0	MFT 单元 0 内输出比较 ch.4 没有中断请求
	1	MFT 单元 0 内输出比较 ch.4 有中断请求
3	0	MFT 单元 0 内输出比较 ch.3 没有中断请求
	1	MFT 单元 0 内输出比较 ch.3 有中断请求
2	0	MFT 单元 0 内输出比较 ch.2 没有中断请求
	1	MFT 单元 0 内输出比较 ch.2 有中断请求
1	0	MFT 单元 0 内输出比较 ch.1 没有中断请求
	1	MFT 单元 0 内输出比较 ch.1 有中断请求
0	0	MFT 单元 0 内输出比较 ch.0 没有中断请求
	1	MFT 单元 0 内输出比较 ch.0 有中断请求

2.21 IRQ31 批量读取寄存器 (IRQ31MON)

IRQ31 批量读取寄存器 (IRQ31MON) 可批量读取分配给中断因素向量 47 的中断请求。

IRQ31MON 指示闪存和基本计时器的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留				FLASHINT	保留										
属性	R				R	R										
初始值	0000				0	00000000000										

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	BTINT															
属性	R															
初始值	0x0000															

寄存器功能

[bit31:28] 保留：保留位

保留位读为 “0”。

[bit27] FLASHINT:

位	描述
0	没有闪存的 RDY、HANG 中断请求
1	有闪存的 RDY、HANG 中断请求

[bit26:16] 保留：保留位

保留位读为 “0”。

[bit15:0] BTINT:

位号	位	描述
15	0	基本计时器 ch.7 上没有 IRQ1 中断请求
	1	基本计时器 ch.7 上有 IRQ1 中断请求
14	0	基本计时器 ch.7 上没有 IRQ0 中断请求
	1	基本计时器 ch.7 上有 IRQ0 中断请求
13	0	基本计时器 ch.6 上没有 IRQ1 中断请求
	1	基本计时器 ch.6 上有 IRQ1 中断请求
12	0	基本计时器 ch.6 上没有 IRQ0 中断请求
	1	基本计时器 ch.6 上有 IRQ0 中断请求
11	0	基本计时器 ch.5 上没有 IRQ1 中断请求
	1	基本计时器 ch.5 上有 IRQ1 中断请求
10	0	基本计时器 ch.5 上没有 IRQ0 中断请求
	1	基本计时器 ch.5 上有 IRQ0 中断请求
9	0	基本计时器 ch.4 上没有 IRQ1 中断请求
	1	基本计时器 ch.4 上有 IRQ1 中断请求
8	0	基本计时器 ch.4 上没有 IRQ0 中断请求
	1	基本计时器 ch.4 上有 IRQ0 中断请求
7	0	基本计时器 ch.3 上没有 IRQ1 中断请求
	1	基本计时器 ch.3 上有 IRQ1 中断请求
6	0	基本计时器 ch.3 上没有 IRQ0 中断请求
	1	基本计时器 ch.3 上有 IRQ0 中断请求
5	0	基本计时器 ch.2 上没有 IRQ1 中断请求
	1	基本计时器 ch.2 上有 IRQ1 中断请求
4	0	基本计时器 ch.2 上没有 IRQ0 中断请求
	1	基本计时器 ch.2 上有 IRQ0 中断请求
3	0	基本计时器 ch.1 上没有 IRQ1 中断请求
	1	基本计时器 ch.1 上有 IRQ1 中断请求
2	0	基本计时器 ch.1 上没有 IRQ0 中断请求
	1	基本计时器 ch.1 上有 IRQ0 中断请求
1	0	基本计时器 ch.0 上没有 IRQ1 中断请求
	1	基本计时器 ch.0 上有 IRQ1 中断请求
0	0	基本计时器 ch.0 上没有 IRQ0 中断请求
	1	基本计时器 ch.0 上有 IRQ0 中断请求

如果通过 DRQSEL 寄存器选择 DMA 传输请求，对应的 BTINT 位为 "0"。

如 Table 2-1 所示，基本计时器中断因素 IRQ0 和 IRQ1 视所用基本计时器功能而变化。

Table 2-1 基本计时器各功能的中断因素

功能	中断因素 IRQ0	中断因素 IRQ1
16 位 PWM 计时器	下溢检测/占空匹配检测	计时器启动触发信号检测
16 位 PPG 计时器	下溢检测	计时器启动触发信号检测
16/32 位重载计时器	下溢检测	计时器启动触发信号检测
16/32 位 PWC 计时器	上溢检测	测量完成检测

3. 使用注意事项

使用中断控制器时应注意以下各点：

- 各外设资源的中断请求信号按等级通知。从中断处理中退出时，总是清除中断请求。
- 分配 NMIX 引脚复用为通用端口。复位释放后的初始值配置为通用端口，而 NMI 输入被屏蔽。使用 NMI 时需要通过配置端口设置使能 NMI。详见“外部中断和 NMI 控制单元”一章。
- 各外设资源的具体事件检测寄存器与中断使能寄存器的相互关系，参见各模块相关章节。

第 7-4 章：中断 (TYPE 1-B)

本章说明 **IRQCMODE=1** 时的异常、中断因素向量及寄存器（**TYPE　1**）。

1. 异常及中断因素向量
2. 寄存器
3. 使用注意事项

代码：9AFIRQC_B-C01.0

1. 异常及中断因素向量

本节列出输入至 NVIC 的异常及中断向量表。

Table 1-1-1 中断因素向量异常及中断

向量号	IRQ 号	异常及中断因素	向量偏移
0	-	栈指针初始值	0x00
1	-	复位	0x04
2	-	不可屏蔽中断 (NMI)/硬件看门狗计时器	0x08
3	-	硬件故障	0x0C
4	-	保留	0x10
5	-	保留	0x14
6	-	保留	0x18
7-10	-	保留	0x1C - 0x2B
11	-	SVCall（监视器调用）	0x2C
12	-	保留	0x30
13	-	保留	0x34
14	-	PendSV	0x38
15	-	SysTick	0x3C
16	0	时钟监视器的异常频率检测 (FCS)	0x40
17	1	软件看门狗计时器	0x44
18	2	低压检测器 (LVD)	0x48
19	3	RCINTSEL0 寄存器内 INTSEL0 位选择的中断源	0x4C
20	4	RCINTSEL0 寄存器内 INTSEL1 位选择的中断源	0x50
21	5	RCINTSEL0 寄存器内 INTSEL2 位选择的中断源	0x54
22	6	RCINTSEL0 寄存器内 INTSEL3 位选择的中断源	0x58
23	7	RCINTSEL1 寄存器内 INTSEL0 位选择的中断源	0x5C
24	8	RCINTSEL1 寄存器内 INTSEL1 位选择的中断源	0x60
25	9	RCINTSEL1 寄存器内 INTSEL2 位选择的中断源	0x64
26	10	RCINTSEL1 寄存器内 INTSEL3 位选择的中断源	0x68
27	11	MFT 单元 0 波形发生器/DTIF（电机紧急停止）/MFS ch.8 的接收中断请求、传输中断请求和状态中断请求	0x6C
28	12	外部引脚中断请求 ch.0 ~ ch.7	0x70
29	13	外部引脚中断请求 ch.8 ~ ch.31	0x74
30	14	双计时器/四位计时器 (QPRC) ch.0	0x78
31	15	MFS ch.0 的接收中断请求、传输中断请求和状态中断请求	0x7C
32	16	MFS ch.1 的接收中断请求、传输中断请求和状态中断请求	0x80
33	17	MFS ch.2 的接收中断请求、传输中断请求和状态中断请求	0x84
34	18	MFS ch.3 的接收中断请求、传输中断请求和状态中断请求	0x88
35	19	MFS ch.4 的接收中断请求	0x8C
36	20	MFS ch.4 的传输中断请求和状态中断请求	0x90

向量号	IRQ 号	异常及中断因素	向量偏移
37	21	MFS ch.5 的接收中断请求	0x94
38	22	MFS ch.5 的传输中断请求和状态中断请求	0x98
39	23	PPG ch.0/2/4/8/10/12/16/18/20	0x9C
40	24	外部主 OSC / 外部副 OSC / 主 PLL / 计时计数器/实时计数器	0xA0
41	25	A/D 转换器单元 0/MFS ch.9 的接收中断请求、传输中断请求和状态中断请求	0xA4
42	26	A/D 转换器单元 1/MFS ch.10 的接收中断请求、传输中断请求和状态中断请求	0xA8
43	27	A/D 转换器单元 2/LCD 控制器/MFS ch.11 的接收中断请求、传输中断请求和状态中断请求	0xAC
44	28	MFT 单元 0 自由运行计时器、输入捕捉、输出比较	0xB0
45	29	MFT 单元 1 自由运行计时器、输入捕捉、输出比较	0xB4
46	30	MFT 单元 2 自由运行计时器、输入捕捉、输出比较/ DMAC ch.0 ~ ch.7	0xB8
47	31	基本计时器 ch.0 ~ ch.7 / 闪存 RDY 中断 / 闪存 HANG 中断	0xBC

可使用 NVIC 内置的系统处理程序优先级寄存器（地址 0xE000ED18、0xE000ED1C、0xE000ED20）配置第 4 至第 15 向量的异常优先级。可使用 NVIC 内置的 IRQ 优先级寄存器（地址 0xE000E400 至 0xE000E42C）配置第 16 及之后向量的外设中断优先级。

使用批量读取寄存器检查第 2、第 16 至第 47 号向量的中断源。有关其他异常及中断，详见《Cortex-M0+ 技术参考手册》中“第 5 章：嵌套向量中断控制器”。

此外，第 2、第 16 至第 47 号向量时，批量读取的中断源可能是多中断因素在各外设宏内被逻辑“或”的信号。详见各外设资源中断的说明。

2. 寄存器

本节说明 DMA 传输请求选择寄存器和中断请求批量读取寄存器。

DMA 传输请求选择寄存器和中断请求批量读取寄存器表

Table 2-1 DMA 传输请求选择寄存器和中断请求批量读取寄存器表

缩写	寄存器名称	参考章节
DRQSEL	DMA 请求选择寄存器	2.1
EXC02MON	EXC02 批量读取寄存器	2.2
IRQ00MON	IRQ00 批量读取寄存器	2.3
IRQ01MON	IRQ01 批量读取寄存器	2.4
IRQ02MON	IRQ02 批量读取寄存器	2.5
IRQ03MON	IRQ03 批量读取寄存器	2.6
IRQ04MON	IRQ04 批量读取寄存器	
IRQ05MON	IRQ05 批量读取寄存器	
IRQ06MON	IRQ06 批量读取寄存器	
IRQ07MON	IRQ07 批量读取寄存器	
IRQ08MON	IRQ08 批量读取寄存器	
IRQ09MON	IRQ09 批量读取寄存器	
IRQ10MON	IRQ10 批量读取寄存器	
IRQ11MON	IRQ11 批量读取寄存器	2.7
IRQ12MON	IRQ12 批量读取寄存器	2.8
IRQ13MON	IRQ13 批量读取寄存器	2.9
IRQ14MON	IRQ14 批量读取寄存器	2.10
IRQ15MON	IRQ15 批量读取寄存器	2.11
IRQ16MON	IRQ16 批量读取寄存器	
IRQ17MON	IRQ17 批量读取寄存器	
IRQ18MON	IRQ18 批量读取寄存器	
IRQ19MON	IRQ19 批量读取寄存器	2.12
IRQ20MON	IRQ20 批量读取寄存器	2.13
IRQ21MON	IRQ21 批量读取寄存器	2.12
IRQ22MON	IRQ22 批量读取寄存器	2.13
IRQ23MON	IRQ23 批量读取寄存器	2.14
IRQ24MON	IRQ24 批量读取寄存器	2.15
IRQ25MON	IRQ25 批量读取寄存器	2.16
IRQ26MON	IRQ26 批量读取寄存器	
IRQ27MON	IRQ27 批量读取寄存器	2.17
IRQ28MON	IRQ28 批量读取寄存器	2.18
IRQ29MON	IRQ29 批量读取寄存器	
IRQ30MON	IRQ30 批量读取寄存器	2.19
IRQ31MON	IRQ31 批量读取寄存器	2.20
IRQCMODE	中断因素向量重定位设置寄存器	2.21
RCINTSEL0	中断因素选择寄存器 0	2.22
RCINTSEL1	中断因素选择寄存器 1	2.23

有关 NVIC 内的寄存器，详见《Cortex-M0+ 技术参考手册》中 “第 5 章：嵌套向量中断控制器”。

2.1 DMA 请求选择寄存器(DRQSEL)

DMA 请求选择寄存器（DRQSEL）选择输出中断信号，是启动 DMA 传输输出到 CPU 的中断请求，还是作为输出到 DMAC 的中断请求。

寄存器配置

位	31	5 4 3 2 1 0
字段	DRQSEL[31:5]	保留
属性	R/W	R
初始值	00000000000000000000000000000000	00000

寄存器功能

[bit31:0] DRQSEL:

位号	位	描述
31	0	外部中断 ch.3 的中断信号作为中断请求输出至 CPU。
	1	外部中断 ch.3 的中断信号作为传输请求输出至 DMAC。
30	0	外部中断 ch.2 的中断信号作为中断请求输出至 CPU。
	1	外部中断 ch.2 的中断信号作为传输请求输出至 DMAC。
29	0	外部中断 ch.1 的中断信号作为中断请求输出至 CPU。
	1	外部中断 ch.1 的中断信号作为传输请求输出至 DMAC。
28	0	外部中断 ch.0 的中断信号作为中断请求输出至 CPU。
	1	外部中断 ch.0 的中断信号作为传输请求输出至 DMAC。
27	0	MFS ch.7 的传输中断信号作为中断请求输出至 CPU。
	1	MFS ch.7 的传输中断信号作为传输请求输出至 DMAC。
26	0	MFS ch.7 的接收中断信号作为中断请求输出至 CPU。
	1	MFS ch.7 的接收中断信号作为传输请求输出至 DMAC。
25	0	MFS ch.6 的传输中断信号作为中断请求输出至 CPU。
	1	MFS ch.6 的传输中断信号作为传输请求输出至 DMAC。
24	0	MFS ch.6 的接收中断信号作为中断请求输出至 CPU。
	1	MFS ch.6 的接收中断信号作为传输请求输出至 DMAC。
23	0	MFS ch.5 的传输中断信号作为中断请求输出至 CPU。
	1	MFS ch.5 的传输中断信号作为传输请求输出至 DMAC。
22	0	MFS ch.5 的接收中断信号作为中断请求输出至 CPU。
	1	MFS ch.5 的接收中断信号作为传输请求输出至 DMAC。
21	0	MFS ch.4 的传输中断信号作为中断请求输出至 CPU。
	1	MFS ch.4 的传输中断信号作为传输请求输出至 DMAC。
20	0	MFS ch.4 的接收中断信号作为中断请求输出至 CPU。
	1	MFS ch.4 的接收中断信号作为传输请求输出至 DMAC。
19	0	MFS ch.3 的传输中断信号作为中断请求输出至 CPU。
	1	MFS ch.3 的传输中断信号作为传输请求输出至 DMAC。
18	0	MFS ch.3 的接收中断信号作为中断请求输出至 CPU。
	1	MFS ch.3 的接收中断信号作为传输请求输出至 DMAC。
17	0	MFS ch.2 的传输中断信号作为中断请求输出至 CPU。
	1	MFS ch.2 的传输中断信号作为传输请求输出至 DMAC。
16	0	MFS ch.2 的接收中断信号作为中断请求输出至 CPU。

位号	位	描述
	1	MFS ch.2 的接收中断信号作为传输请求输出至 DMAC。
15	0	MFS ch.1 的传输中断信号作为中断请求输出至 CPU。
	1	MFS ch.1 的传输中断信号作为传输请求输出至 DMAC。
14	0	MFS ch.1 的接收中断信号作为中断请求输出至 CPU。
	1	MFS ch.1 的接收中断信号作为传输请求输出至 DMAC。
13	0	MFS ch.0 的传输中断信号作为中断请求输出至 CPU。
	1	MFS ch.0 的传输中断信号作为传输请求输出至 DMAC。
12	0	MFS ch.0 的接收中断信号作为中断请求输出至 CPU。
	1	MFS ch.0 的接收中断信号作为传输请求输出至 DMAC。
11	0	基本计时器 ch.6 的 IRQ0 中断信号作为中断请求输出至 CPU 。
	1	基本计时器 ch.6 的 IRQ0 中断信号作为传输请求输出至 DMAC（包括扩展）。
10	0	基本计时器 ch.4 的 IRQ0 中断信号作为中断请求输出至 CPU 。
	1	基本计时器 ch.4 的 IRQ0 中断信号作为传输请求输出至 DMAC（包括扩展）。
9	0	基本计时器 ch.2 的 IRQ0 中断信号作为中断请求输出至 CPU。
	1	基本计时器 ch.2 的 IRQ0 中断信号作为传输请求输出至 DMAC。
8	0	基本计时器 ch.0 的 IRQ0 中断信号作为中断请求输出至 CPU 。
	1	基本计时器 ch.0 的 IRQ0 中断信号作为传输请求输出至 DMAC。
7	0	A/D 转换器单元 2 的扫描转换中断信号作为中断请求输出至 CPU。
	1	A/D 转换器单元 2 的扫描转换中断信号作为中断请求输出至 DMAC。
6	0	A/D 转换器单元 1 的扫描转换中断信号作为中断请求输出至 CPU。
	1	A/D 转换器单元 1 的扫描转换中断信号作为中断请求输出至 DMAC。
5	0	A/D 转换器单元 0 的扫描转换中断信号作为中断请求输出至 CPU。
	1	A/D 转换器单元 0 的扫描转换中断信号作为中断请求输出至 DMAC。

MFS: 多功能串口

[bit4:0] 保留：保留位

保留位读为 “0”。

注意事项：

- *若选择中断信号为传输请求发送至 DMAC，无论是否发生中断，相关中断请求批量读取寄存器（IRQxxMON, xx = 00 ~ 31）的读取位值变为 “0”。*
- *改变 DRQSEL 设置时，应在更改前清除相关外围设备的中断请求信号。*
- *如果不是 DRQSEL 设置所指定的中断信号，则 DMA 传输不能通过硬件启动。*

2.2 EXC02 批量读取寄存器 (EXC02MON)

EXC02 批量读取寄存器 (EXC02MON) 可批量读取分配给中断因素向量 2 的中断请求。

EXC02MON 指示硬件看门狗计时器和 NMIX 外部引脚的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留														HWINT	NMI
属性	R														R	R
初始值	00000000000000														0	0

寄存器功能

[bit31:2] 保留：保留位

保留位读为 “0”。

[bit1] HWINT:

位	描述
0	没有硬件看门狗计时器中断请求
1	有硬件看门狗计时器中断请求

[bit0] NMI:

位	描述
0	没有 NMIX 外部引脚中断请求
1	有 NMIX 外部引脚中断请求

2.3 IRQ00 批量读取寄存器(IRQ00MON)

IRQ00 批量读取寄存器 (IRQ00MON) 可批量读取分配给中断因素向量 16 的中断请求。

IRQ00MON 指示通过 CSV 进行的异常频率检测的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留															FCSINT
属性	R															R
初始值	000000000000000															0

寄存器功能

[bit31:1] 保留：保留位

保留位读为 “0”。

[bit0] FCSINT:

位	描述
0	没有通过 CSV 中断请求进行的异常频率检测
1	有通过 CSV 中断请求进行的异常频率检测

2.4 IRQ01 批量读取寄存器(IRQ01MON)

IRQ01 批量读取寄存器 (IRQ01MON) 可批量读取分配给中断因素向量 17 的中断请求。

IRQ01MON 指示软件看门狗计时器的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留															SWWDTINT
属性	R															R
初始值	000000000000000															0

寄存器功能

[bit31:1] 保留：保留位

保留位读为 “0”。

[bit0] SWWDTINT:

位	描述
0	没有软件看门狗计时器中断请求
1	有软件看门狗计时器中断请求

2.5 IRQ02 批量读取寄存器 (IRQ02MON)

IRQ02 批量读取寄存器 (IRQ02MON) 可批量读取分配给中断因素向量 18 的中断请求。

IRQ02MON 指示低压检测 (LVD) 的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留															LVDINT
属性	R															R
初始值	000000000000000															0

寄存器功能

[bit31:1] 保留：保留位

保留位读为 “0”。

[bit0] LVDINT:

位	描述
0	没有低压检测 (LVD) 中断请求
1	有低压检测 (LVD) 中断请求

2.6 IRQ03~IRQ10 批量读取寄存器 (IRQ03MON ~ IRQ10MON)

IRQ03 批量读取寄存器 (IRQ03MON) 可批量读取分配给中断因素向量 19 的中断请求。

IRQ04 批量读取寄存器 (IRQ04MON) 可批量读取分配给中断因素向量 20 的中断请求。

IRQ05 批量读取寄存器 (IRQ05MON) 可批量读取分配给中断因素向量 21 的中断请求。

IRQ06 批量读取寄存器 (IRQ06MON) 可批量读取分配给中断因素向量 22 的中断请求。

IRQ07 批量读取寄存器 (IRQ07MON) 可批量读取分配给中断因素向量 23 的中断请求。

IRQ08 批量读取寄存器 (IRQ08MON) 可批量读取分配给中断因素向量 24 的中断请求。

IRQ09 批量读取寄存器 (IRQ09MON) 可批量读取分配给中断因素向量 25 的中断请求。

IRQ10 批量读取寄存器 (IRQ10MON) 可批量读取分配给中断因素向量 26 的中断请求。

IRQ03MON 指示 RCINTSEL0 寄存器内 INTSEL0 位选择的中断请求状态。

IRQ04MON 指示 RCINTSEL0 寄存器内 INTSEL1 位选择的中断请求状态。

IRQ05MON 指示 RCINTSEL0 寄存器内 INTSEL2 位选择的中断请求状态。

IRQ06MON 指示 RCINTSEL0 寄存器内 INTSEL3 位选择的中断请求状态。

IRQ07MON 指示 RCINTSEL1 寄存器内 INTSEL0 位选择的中断请求状态。

IRQ08MON 指示 RCINTSEL1 寄存器内 INTSEL1 位选择的中断请求状态。

IRQ09MON 指示 RCINTSEL1 寄存器内 INTSEL2 位选择的中断请求状态。

IRQ10MON 指示 RCINTSEL1 寄存器内 INTSEL3 位选择的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留															RCINT
属性	R															R
初始值	00000000000000															0

寄存器功能

[bit31:1] 保留：保留位

保留位读为 “0”。

[bit0] RCINT:

位	描述
0	相关 RCINTSEL0:INTSELx/RCINTSEL1:INTSELx 没有选择中断请求
1	相关 RCINTSEL0:NTSELx/RCINTSEL1:INTSELx 选择有中断请求*

*: 若选择基本计时器为中断因素，通过中断因素 IRQ0 或 IRQ1 将该位设置为 "1"。

2.7 IRQ11 批量读取寄存器 (IRQxxMON)

IRQ11 批量读取寄存器 (IRQ11MON) 可批量读取分配给中断因素向量 27 的中断请求。

IRQ11MON 指示 MFT 单元 0 和 MFS ch.8 的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留									MFSINT			WAVEINT			
属性	R									R			R			
初始值	000000000									000			0000			

寄存器功能

[bit31:7] 保留：保留位

保留位读为 “0”。

[bit6:4] MFSINT:

位号	位	描述
6	0	没有对应 MFS 通道的状态中断请求
	1	有对应 MFS 通道的状态中断请求
5	0	没有对应 MFS 通道的传输中断请求
	1	有对应 MFS 通道的传输中断请求
4	0	没有对应 MFS 通道的接收中断请求
	1	有对应 MFS 通道的接收中断请求

[bit3:0] WAVEINT:

位号	位	描述
3	0	没有对应的 MFT 单元内 WFG 计时器 54 的中断请求
	1	有对应的 MFT 单元内 WFG 计时器 54 的中断请求
2	0	没有对应的 MFT 单元内 WFG 计时器 32 的中断请求
	1	有对应的 MFT 单元内 WFG 计时器 32 的中断请求
1	0	没有对应的 MFT 单元内 WFG 计时器 10 的中断请求
	1	有对应的 MFT 单元内 WFG 计时器 10 的中断请求
0	0	没有对应 MFT 单元内 DTIF (电机紧急停止)的中断请求
	1	有对应 MFT 单元内 DTIF (电机紧急停止)的中断请求

2.8 IRQ12 批量读取寄存器 (IRQ12MON)

IRQ12 批量读取寄存器 (IRQ12MON) 可批量读取分配给中断因素向量 28 的中断请求。

IRQ12MON 指示外部中断 ch.0 ~ ch.7 的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留								EXTINT							
属性	R								R							
初始值	0x00								0x00							

寄存器功能

[bit31:8] 保留：保留位

保留位读为 “0”。

[bit7:0] EXTINT:

位号	位	描述
7	0	没有外部中断 ch.7 的中断请求
	1	有外部中断 ch.7 的中断请求
6	0	没有外部中断 ch.6 的中断请求
	1	有外部中断 ch.6 的中断请求
5	0	没有外部中断 ch.5 的中断请求
	1	有外部中断 ch.5 的中断请求
4	0	没有外部中断 ch.4 的中断请求
	1	有外部中断 ch.4 的中断请求
3	0	没有外部中断 ch.3 的中断请求
	1	有外部中断 ch.3 的中断请求
2	0	没有外部中断 ch.2 的中断请求
	1	有外部中断 ch.2 的中断请求
1	0	没有外部中断 ch.1 的中断请求
	1	有外部中断 ch.1 的中断请求
0	0	没有外部中断 ch.0 的中断请求
	1	有外部中断 ch.0 的中断请求

若通过 DRQSEL 寄存器选择 DMA 传输请求，对应的 EXTINT 位为 "0"。

2.9 IRQ13 批量读取寄存器 (IRQ13MON)

IRQ13 批量读取寄存器 (IRQ13MON) 可批量读取分配给中断因素向量 29 的中断请求。

IRQ13MON 指示外部中断 ch.8 ~ ch.31 的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留								EXTINT							
属性	R								R							
初始值	0x00								0x00							

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	EXTINT															
属性	R															
初始值	0x0000															

寄存器功能

[bit31:24] 保留：保留位

保留位读为 “0”。

[bit23:0] EXTINT:

位号	位	描述
23	0	没有外部中断 ch.31 的中断请求
	1	有外部中断 ch.31 的中断请求
22	0	没有外部中断 ch.30 的中断请求
	1	有外部中断 ch.30 的中断请求
21	0	没有外部中断 ch.29 的中断请求
	1	有外部中断 ch.29 的中断请求
20	0	没有外部中断 ch.28 的中断请求
	1	有外部中断 ch.28 的中断请求
19	0	没有外部中断 ch.27 的中断请求
	1	有外部中断 ch.27 的中断请求
18	0	没有外部中断 ch.26 的中断请求
	1	有外部中断 ch.26 的中断请求
17	0	没有外部中断 ch.25 的中断请求
	1	有外部中断 ch.25 的中断请求
16	0	没有外部中断 ch.24 的中断请求
	1	有外部中断 ch.24 的中断请求
15	0	没有外部中断 ch.23 的中断请求
	1	有外部中断 ch.23 的中断请求
14	0	没有外部中断 ch.22 的中断请求
	1	有外部中断 ch.22 的中断请求
13	0	没有外部中断 ch.21 的中断请求
	1	有外部中断 ch.21 的中断请求
12	0	没有外部中断 ch.20 的中断请求

位号	位	描述
	1	有外部中断 ch.20 的中断请求
11	0	没有外部中断 ch.19 的中断请求
	1	有外部中断 ch.19 的中断请求
10	0	没有外部中断 ch.18 的中断请求
	1	有外部中断 ch.18 的中断请求
9	0	没有外部中断 ch.17 的中断请求
	1	有外部中断 ch.17 的中断请求
8	0	没有外部中断 ch.16 的中断请求
	1	有外部中断 ch.16 的中断请求
7	0	没有外部中断 ch.15 的中断请求
	1	有外部中断 ch.15 的中断请求
6	0	没有外部中断 ch.14 的中断请求
	1	有外部中断 ch.14 的中断请求
5	0	没有外部中断 ch.13 的中断请求
	1	有外部中断 ch.13 的中断请求
4	0	没有外部中断 ch.12 的中断请求
	1	有外部中断 ch.12 的中断请求
3	0	没有外部中断 ch.11 的中断请求
	1	有外部中断 ch.11 的中断请求
2	0	没有外部中断 ch.10 的中断请求
	1	有外部中断 ch.10 的中断请求
1	0	没有外部中断 ch.9 的中断请求
	1	有外部中断 ch.9 的中断请求
0	0	没有外部中断 ch.8 的中断请求
	1	有外部中断 ch.8 的中断请求

2.10 IRQ14 批量读取寄存器 (IRQ14MON)

IRQ14 批量读取寄存器 (IRQ14MON) 批量读取分配给中断因素向量 30 的中断请求。

IRQ14MON 指示双计时器和 QPRC 的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留								QUDINT						TIMINT	
属性	R								R						R	
初始值	0x00								000000						00	

寄存器功能

[bit31:8] 保留：保留位

保留位读为 “0”。

[bit7:2] QUDINT:

位号	位	描述
7	0	没有 QPRC ch.0 的 PC 匹配和 RC 匹配中断请求
	1	有 QPRC ch.0 的 PC 匹配和 RC 匹配中断请求
6	0	QPRC ch.0 没有检测到范围外 RC 的中断请求
	1	QPRC ch.0 检测到范围外 RC 的中断请求
5	0	没有 QPRC ch.0 的 PC 计数反向中断请求
	1	有 QPRC ch.0 的 PC 计数反向中断请求
4	0	没有 QPRC ch.0 的上溢/下溢/零索引中断请求
	1	有 QPRC ch.0 的上溢/下溢/零索引中断请求
3	0	没有 QPRC ch.0 的 PC 和 RC 匹配中断请求
	1	有 QPRC ch.0 的 PC 和 RC 匹配中断请求
2	0	没有 QPRC ch.0 的 PC 匹配中断请求
	1	有 QPRC ch.0 的 PC 匹配中断请求

[bit1:0] TIMINT:

位号	位	描述
1	0	没有双计时器 TIMINT2 中断请求
	1	有双计时器 TIMINT2 中断请求
0	0	没有双计时器 TIMINT1 中断请求
	1	有双计时器 TIMINT1 中断请求

2.11 IRQ15 ~ IRQ18 批量读取寄存器 (IRQ15MON ~ IRQ18MON)

IRQ15 批量读取寄存器 (IRQ15MON) 可批量读取分配给中断因素向量 31 的中断请求。

IRQ16 批量读取寄存器 (IRQ16MON) 可批量读取分配给中断因素向量 32 的中断请求。

IRQ17 批量读取寄存器 (IRQ17MON) 批量读取分配给中断因素向量 33 的中断请求。

IRQ18 批量读取寄存器 (IRQ18MON) 可批量读取分配给中断因素向量 34 的中断请求。

IRQ15MON 指示 MFS ch.0 的接收中断请求、传输中断请求及状态请求的状态。

IRQ16MON 指示 FS ch.1 的接收中断请求、传输中断请求及状态请求的状态。

IRQ17MON 指示 MFS ch.2 的接收中断请求、传输中断请求及状态请求的状态。

IRQ18MON 指示 MFS ch.3 的接收中断请求、传输中断请求及状态请求的状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留													MFSINT		
属性	R													R		
初始值	0000000000000													000		

寄存器功能

[bit31:3] 保留：保留位

保留位读为 “0”。

[bit2:0] MFSINT:

位号	位	描述
2	0	没有对应 MFS 通道的状态中断请求
	1	有对应 MFS 通道的状态中断请求
1	0	没有对应 MFS 通道的传输中断请求
	1	有对应 MFS 通道的传输中断请求
0	0	没有对应 MFS 通道的接收中断请求
	1	有对应 MFS 通道的接收中断请求

如果 DRQSEL 寄存器已选择 DMA 传输请求，对应的 MFS 通道的 MFSINT 位为 "0"。

2.12 IRQ19/21 批量读取寄存器 (IRQxxMON)

IRQ19 批量读取寄存器 (IRQ19MON) 可批量读取分配给中断因素向量 35 的中断请求。

IRQ21 批量读取寄存器 (IRQ21MON) 可批量读取分配给中断因素向量 37 的中断请求。

IRQ19MON 指示 MFS ch.4.的接收中断请求状态。

IRQ21MON 指示 MFS ch.5. 的接收中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留															MFSINT
属性	R															R
初始值	000000000000000															0

寄存器功能

[bit31:1] 保留：保留位

保留位读为 “0”。

[bit0] MFSINT:

位号	位	描述
0	0	没有对应 MFS 通道的接收中断请求
	1	有对应 MFS 通道的接收中断请求

如果 DRQSEL 寄存器已选择 DMA 传输请求，对应的 MFS 通道的 MFSINT 位为 "0"。

2.13 IRQ20/22 批量读取寄存器 (IRQxxMON)

IRQ20 批量读取寄存器 (IRQ20MON) 批量读取分配给中断因素向量 36 的中断请求。

IRQ22 批量读取寄存器 (IRQ22MON) 批量读取分配给中断因素向量 38 的中断请求。

IRQ20MON 指示 MFS ch.4 的传输中断请求及状态中断请求的状态。

IRQ22MON 指示 MFS ch.5 的传输中断请求及状态中断请求的状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留														MFSINT	
属性	R														R	
初始值	00000000000000														00	

寄存器功能

[bit31:2] 保留：保留位

保留位读为 “0”。

[bit1:0] MFSINT:

位号	位	描述
1	0	没有对应 MFS 通道的状态中断请求
	1	有对应 MFS 通道的状态中断请求
0	0	没有对应 MFS 通道的传输中断请求
	1	有对应 MFS 通道的传输中断请求

如果 DRQSEL 寄存器已选择 DMA 传输请求，对应的 MFS 通道的 MFSINT 位为 "0"。

2.14 IRQ23 批量读取寄存器 (IRQ23MON)

IRQ23 批量读取寄存器 (IRQ23MON) 批量读取分配给中断因素向量 39 的中断请求。

IRQ23MON 指示 PPG 的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留							PPGINT								
属性	R							R								
初始值	0000000							000000000								

寄存器功能

[bit31:9] 保留：保留位

保留位读为 “0”。

[bit8:0] PPGINT:

位号	位	描述
8	0	没有 PPG ch.20 的中断请求
	1	有 PPG ch.20 的中断请求
7	0	没有 PPG ch.18 的中断请求
	1	有 PPG ch.18 的中断请求
6	0	没有 PPG ch.16 的中断请求
	1	有 PPG ch.16 的中断请求
5	0	没有 PPG ch.12 的中断请求
	1	有 PPG ch.12 的中断请求
4	0	没有 PPG ch.10 的中断请求
	1	有 PPG ch.10 的中断请求
3	0	没有 PPG ch.8 的中断请求
	1	有 PPG ch.8 的中断请求
2	0	没有 PPG ch.4 的中断请求
	1	有 PPG ch.4 的中断请求
1	0	没有 PPG ch.2 的中断请求
	1	有 PPG ch.2 的中断请求
0	0	没有 PPG ch.0 的中断请求
	1	有 PPG ch.0 的中断请求

2.15 IRQ24 批量读取寄存器 (IRQ24MON)

IRQ24 批量读取寄存器 (IRQ24MON) 批量读取分配给中断因素向量 40 的中断请求。

1. IRQ24MON 指示 RTC、计时计数器、主 PLL 振荡、副振荡和主时钟振荡的中断请求状态。

寄存器配置

位	31							8
字段	保留							
属性	R							
初始值	0x000000							

位	7	6	5	4	3	2	1	0
字段	保留		RTCINT	WCINT	保留	MPLLINT	SOSCINT	MOSCINT
属性	R		R	R	R	R	R	R
初始值	00		0	0	0	0	0	0

寄存器功能

[bit31:6] 保留：保留位
保留位读为 “0”。

[bit5] RTCINT:

位	描述
0	没有 RTC 中断请求
1	有 RTC 中断请求

[bit4] WCINT:

位	描述
0	没有计时计数器中断请求
1	有计时计数器中断请求

[bit3] 保留：保留位
保留位读为 “0”。

[bit2] MPLLINT:

位	描述
0	没有主 PLL 振荡的稳定等待完成中断请求
1	有主 PLL 振荡的稳定等待完成中断请求

[bit1] SOSCINT:

位	描述
0	没有副时钟振荡的稳定等待完成中断请求
1	有副时钟振荡的稳定等待完成中断请求

[bit0] MOSCINT:

位	描述
0	没有主时钟振荡的稳定等待完成中断请求
1	有主时钟振荡的稳定等待完成中断请求

2.16 IRQ25/26 批量读取寄存器(IRQxxMON)

IRQ25 批量读取寄存器 (IRQ25MON) 批量读取分配给中断因素向量 41 的中断请求。

IRQ26 批量读取寄存器 (IRQ26MON) 批量读取分配给中断因素向量 42 的中断请求。

IRQ25MON 指示 A/D 转换器单元 0 和 MFS ch.9 的中断请求状态。

IRQ26MON 指示 A/D 转换器单元 1 和 MFS ch.10 的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留								MFSINT			ADCINT				
属性	R								R			R				
初始值	0x00								000			00000				

寄存器功能

[bit31:8] 保留：保留位

保留位读为 “0”。

[bit7:5] MFSINT:

位号	位	描述
7	0	没有对应 MFS 通道的状态中断请求
	1	有对应 MFS 通道的状态中断请求
6	0	没有对应 MFS 通道的传输中断请求
	1	有对应 MFS 通道的传输中断请求
5	0	没有对应 MFS 通道的接收中断请求
	1	有对应 MFS 通道的接收中断请求

[bit4:0] ADCINT:

位号	位	描述
4	0	没有对应的 A/D 转换器单元的范围比较结果中断请求
	1	有对应的 A/D 转换器单元的范围比较结果中断请求
3	0	没有对应的 A/D 转换器单元的转换结果比较中断请求
	1	有对应的 A/D 转换器单元的转换结果比较中断请求
2	0	没有对应的 A/D 转换器单元的 FIFO 溢出中断请求
	1	有对应的 A/D 转换器单元的 FIFO 溢出中断请求
1	0	没有对应的 A/D 转换器单元的扫描转换中断请求
	1	有对应的 A/D 转换器单元的扫描转换中断请求

位号	位	描述
0	0	没有对应的 A/D 转换器单元的优先转换中断请求
	1	有对应的 A/D 转换器单元的优先转换中断请求

如果 DRQSEL 寄存器已选择 DMA 传输请求，对应的 ADCINT 位为 "0"。

2.17 IRQ27 批量读取寄存器(IRQ27MON)

IRQ27 批量读取寄存器 (IRQ27MON) 批量读取分配给中断因素向量 43 的中断请求。

IRQ27MON 指示 A/D 转换器单元 2、LCD 控制器和 MFS ch.11 的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留							MFSINT			LCDCINT	ADCINT				
属性	R							R			R	R				
初始值	0000000							000			0	00000				

寄存器功能

[bit31:9] 保留：保留位

保留位读为 “0”。

[bit8:6] MFSINT:

位号	位	描述
8	0	没有 MFS ch.11 的状态中断请求
	1	有对应 MFS ch.11 的状态中断请求
7	0	没有对应 MFS ch.11 的传输中断请求
	1	有对应 MFS ch.11 的传输中断请求
6	0	没有对应 MFS ch.11 的接收中断请求
	1	有对应 MFS ch.11 的接收中断请求

[bit5] LCDCINT:

位	描述
0	没有 LCD 控制器的中断请求
1	有 LCD 控制器的中断请求

[bit4:0] ADCINT:

位号	位	描述
4	0	没有对应的 A/D 转换器单元 2 的范围比较结果中断请求
	1	有对应的 A/D 转换器单元 2 的范围比较结果中断请求
3	0	没有 A/D 转换器单元 2 的转换结果比较中断请求
	1	有 A/D 转换器单元 2 的转换结果比较中断请求
2	0	没有 A/D 转换器单元 2 的 FIFO 溢出中断请求
	1	有 A/D 转换器单元 2 的 FIFO 溢出中断请求

位号	位	描述
1	0	没有 A/D 转换器单元 2 的扫描转换中断请求
	1	有 A/D 转换器单元 2 的扫描转换中断请求
0	0	没有 A/D 转换器单元 2 的优先转换中断请求
	1	有 A/D 转换器单元 2 的优先转换中断请求

如果 DRQSEL 寄存器已选择 DMA 传输请求，对应的 ADCINT 位为 "0"。

2.18 IRQ28/29 批量读取寄存器 (IRQxxMON)

IRQ28 批量读取寄存器 (IRQ28MON) 批量读取分配给中断因素向量 44 的中断请求。

IRQ29 批量读取寄存器 (IRQ29MON) 批量读取分配给中断因素向量 45 的中断请求。

IRQ28MON 指示 MFT 单元 0 的中断请求状态。

IRQ29MON 指示 MFT 单元 1 的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	OCUINT						ICUINT				FRTINT					
属性	R						R				R					
初始值	000000						0000				000000					

寄存器功能

[bit31:16] 保留：保留位

保留位读为 “0”。

[bit15:10] OCUINT:

位号	位	描述
15	0	对应 MFT 单元内输出比较 ch.5 没有中断请求
	1	对应 MFT 单元内输出比较 ch.5 有中断请求
14	0	对应 MFT 单元内输出比较 ch.4 没有中断请求
	1	对应 MFT 单元内输出比较 ch.4 有中断请求
13	0	对应 MFT 单元内输出比较 ch.3 没有中断请求
	1	对应 MFT 单元内输出比较 ch.3 有中断请求
12	0	对应 MFT 单元内输出比较 ch.2 没有中断请求
	1	对应 MFT 单元内输出比较 ch.2 有中断请求
11	0	对应 MFT 单元内输出比较 ch.1 没有中断请求
	1	对应 MFT 单元内输出比较 ch.1 有中断请求
10	0	对应 MFT 单元内输出比较 ch.0 没有中断请求
	1	对应 MFT 单元内输出比较 ch.0 有中断请求

[bit9:6] ICUINT:

位号	位	描述
9	0	对应 MFT 单元内输入捕捉 ch.3 没有中断请求
	1	对应 MFT 单元内输入捕捉 ch.3 有中断请求

位号	位	描述
8	0	对应 MFT 单元内输入捕捉 ch.2 没有中断请求
	1	对应 MFT 单元内输入捕捉 ch.2 有中断请求
7	0	对应 MFT 单元内输入捕捉 ch.1 没有中断请求
	1	对应 MFT 单元内输入捕捉 ch.1 有中断请求
6	0	对应 MFT 单元内输入捕捉 ch.0 没有中断请求
	1	对应 MFT 单元内输入捕捉 ch.0 有中断请求

[bit5:0] FRTINT:

位号	位	描述
5	0	对应 MFT 单元内自由运行计时器 ch.2 没有零检测中断请求
	1	对应 MFT 单元内自由运行计时器 ch.2 有零检测中断请求
4	0	对应 MFT 单元内自由运行计时器 ch.1 没有零检测中断请求
	1	对应 MFT 单元内自由运行计时器 ch.1 有零检测中断请求
3	0	对应 MFT 单元内自由运行计时器 ch.0 没有零检测中断请求
	1	对应 MFT 单元内自由运行计时器 ch.0 有零检测中断请求
2	0	对应 MFT 单元内自由运行计时器 ch.2 没有峰值检测中断请求
	1	对应 MFT 单元内自由运行计时器 ch.2 有峰值检测中断请求
1	0	对应 MFT 单元内自由运行计时器 ch.1 没有峰值检测中断请求
	1	对应 MFT 单元内自由运行计时器 ch.1 有峰值检测中断请求
0	0	对应 MFT 单元内自由运行计时器 ch.0 没有峰值检测中断请求
	1	对应 MFT 单元内自由运行计时器 ch.0 有峰值检测中断请求

2.19 IRQ30 批量读取寄存器 (IRQ30MON)

IRQ30 批量读取寄存器 (IRQ30MON) 批量读取分配给中断因素向量 46 的中断请求。

IRQ30MON 指示 MFT 单元 2 及 DMAC ch.0 ~ ch.7 的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留								DMAINT							
属性	R								R							
初始值	0x00								0x00							

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	OCUINT						ICUINT				FRTINT					
属性	R						R				R					
初始值	000000						0000				000000					

寄存器功能

[bit31:24] 保留：保留位

保留位读为 “0”。

[bit23:16] DMAINT:

位号	位	描述
23	0	DMA 控制器 ch.7 没有中断请求
	1	DMA 控制器 ch.7 有中断请求
22	0	DMA 控制器 ch.6 没有中断请求
	1	DMA 控制器 ch.6 有中断请求
21	0	DMA 控制器 ch.5 没有中断请求
	1	DMA 控制器 ch.5 有中断请求
20	0	DMA 控制器 ch.4 没有中断请求
	1	DMA 控制器 ch.4 有中断请求
19	0	DMA 控制器 ch.3 没有中断请求
	1	DMA 控制器 ch.3 有中断请求
18	0	DMA 控制器 ch.2 没有中断请求
	1	DMA 控制器 ch.2 有中断请求
17	0	DMA 控制器 ch.1 没有中断请求
	1	DMA 控制器 ch.1 有中断请求
16	0	DMA 控制器 ch.0 没有中断请求
	1	DMA 控制器 ch.0 有中断请求

[bit15:10] OCUINT:

位号	位	描述
15	0	对应 MFT 单元内输出比较 ch.5 没有中断请求
	1	对应 MFT 单元内输出比较 ch.5 有中断请求
14	0	对应 MFT 单元内输出比较 ch.4 没有中断请求
	1	对应 MFT 单元内输出比较 ch.4 有中断请求
13	0	对应 MFT 单元内输出比较 ch.3 没有中断请求
	1	对应 MFT 单元内输出比较 ch.3 有中断请求
12	0	对应 MFT 单元内输出比较 ch.2 没有中断请求
	1	对应 MFT 单元内输出比较 ch.2 有中断请求
11	0	对应 MFT 单元内输出比较 ch.1 没有中断请求
	1	对应 MFT 单元内输出比较 ch.1 有中断请求
10	0	对应 MFT 单元内输出比较 ch.0 没有中断请求
	1	对应 MFT 单元内输出比较 ch.0 有中断请求

[bit9:6] ICUINT:

位号	位	描述
9	0	对应 MFT 单元内输入捕捉 ch.3 没有中断请求
	1	对应 MFT 单元内输入捕捉 ch.3 有中断请求
8	0	对应 MFT 单元内输入捕捉 ch.2 没有中断请求
	1	对应 MFT 单元内输入捕捉 ch.2 有中断请求
7	0	对应 MFT 单元内输入捕捉 ch.1 没有中断请求
	1	对应 MFT 单元内输入捕捉 ch.1 有中断请求
6	0	对应 MFT 单元内输入捕捉 ch.0 没有中断请求
	1	对应 MFT 单元内输入捕捉 ch.0 有中断请求

[bit5:0] FRTINT:

位号	位	描述
5	0	对应 MFT 单元内自由运行计时器 ch.2 没有零检测中断请求
	1	对应 MFT 单元内自由运行计时器 ch.2 有零检测中断请求
4	0	对应 MFT 单元内自由运行计时器 ch.1 没有零检测中断请求
	1	对应 MFT 单元内自由运行计时器 ch.1 有零检测中断请求
3	0	对应 MFT 单元内自由运行计时器 ch.0 没有零检测中断请求
	1	对应 MFT 单元内自由运行计时器 ch.0 有零检测中断请求
2	0	对应 MFT 单元内自由运行计时器 ch.2 没有峰值检测中断请求
	1	对应 MFT 单元内自由运行计时器 ch.2 有峰值检测中断请求
1	0	对应 MFT 单元内自由运行计时器 ch.1 没有峰值检测中断请求
	1	对应 MFT 单元内自由运行计时器 ch.1 有峰值检测中断请求
0	0	对应 MFT 单元内自由运行计时器 ch.0 没有峰值检测中断请求
	1	对应 MFT 单元内自由运行计时器 ch.0 有峰值检测中断请求

2.20 IRQ31 批量读取寄存器 (IRQ31MON)

IRQ31 批量读取寄存器 (IRQ31MON) 批量读取分配给中断因素向量 47 的中断请求。

IRQ31MON 指示闪存及基本计时器的中断请求状态。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留				FLASHINT	保留										
属性	R				R	R										
初始值	0000				0	00000000000										

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	BTINT															
属性	R															
初始值	0x0000															

寄存器功能

[bit31:28] 保留：保留位

保留位读为 “0”。

[bit27] FLASHINT:

位	描述
0	没有闪存的 RDY 和 HANG 中断请求
1	有闪存的 RDY 和 HANG 中断请求

[bit26:16] 保留：保留位

保留位读为 “0”。

[bit15:0] BTINT:

位号	位	描述
15	0	基本计时器 ch.7 上没有 IRQ1 中断请求
	1	基本计时器 ch.7 上有 IRQ1 中断请求
14	0	基本计时器 ch.7 上没有 IRQ0 中断请求
	1	基本计时器 ch.7 上有 IRQ0 中断请求
13	0	基本计时器 ch.6 上没有 IRQ1 中断请求
	1	基本计时器 ch.6 上有 IRQ1 中断请求
12	0	基本计时器 ch.6 上没有 IRQ0 中断请求
	1	基本计时器 ch.6 上有 IRQ0 中断请求
11	0	基本计时器 ch.5 上没有 IRQ1 中断请求
	1	基本计时器 ch.5 上有 IRQ1 中断请求
10	0	基本计时器 ch.5 上没有 IRQ0 中断请求
	1	基本计时器 ch.5 上有 IRQ0 中断请求

位号	位	描述
9	0	基本计时器 ch.4 上没有 IRQ1 中断请求
	1	基本计时器 ch.4 上有 IRQ1 中断请求
8	0	基本计时器 ch.4 上没有 IRQ0 中断请求
	1	基本计时器 ch.4 上有 IRQ0 中断请求
7	0	基本计时器 ch.3 上没有 IRQ1 中断请求
	1	基本计时器 ch.3 上有 IRQ1 中断请求
6	0	基本计时器 ch.3 上没有 IRQ0 中断请求
	1	基本计时器 ch.3 上有 IRQ0 中断请求
5	0	基本计时器 ch.2 上没有 IRQ1 中断请求
	1	基本计时器 ch.2 上有 IRQ1 中断请求
4	0	基本计时器 ch.2 上没有 IRQ0 中断请求
	1	基本计时器 ch.2 上有 IRQ0 中断请求
3	0	基本计时器 ch.1 上没有 IRQ1 中断请求
	1	基本计时器 ch.1 上有 IRQ1 中断请求
2	0	基本计时器 ch.1 上没有 IRQ0 中断请求
	1	基本计时器 ch.1 上有 IRQ0 中断请求
1	0	基本计时器 ch.0 上没有 IRQ1 中断请求
	1	基本计时器 ch.0 上有 IRQ1 中断请求
0	0	基本计时器 ch.0 上没有 IRQ0 中断请求
	1	基本计时器 ch.0 上有 IRQ0 中断请求

如果 DRQSEL 寄存器已选择 DMA 传输请求，对应的 BTINT 位为 "0"。

如 Table 2-2 所示，基本计时器中断因素 IRQ0 和 IRQ1 随所用基本计时器功能而变化。

Table 2-2 各项基本计时器功能的中断因素

功能	中断因素 IRQ0	中断因素 IRQ1
16 位 PWM 计时器	下溢检测/占空匹配检测	计时器启动触发器检测
16 位 PPG 计时器	下溢检测	计时器启动触发器检测
16/32 位重载计时器	下溢检测	计时器启动触发器检测
16/32 位 PWC 计时器	上溢检测	测量完成检测

2.21 中断因素向量重定位设置寄存器 (IRQCMODE)

中断因素向量重定位设置寄存器 (IRQCMODE) 选择根据“中断 (A)”章 Table 1-1 或根据本章 Table 1-1-1 分配中断因素向量。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R/W															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留															IRQCMODE
属性	R/W															R/W
初始值	000000000000000															0

寄存器功能

[bit31:1] 保留：保留位

保留位读为 “0”。

[bit0] IRQCMODE:

位	描述
0	根据 “中断 (A)”一章的 Table 1-1-1 分配中断因素向量。
1	根据本章 Table 1-1-1 分配中断因素向量。

2.22 中断因素选择寄存器 0 (RCINTSEL0)

中断因素选择寄存器 0 (RCINTSEL0) 选择中断向量 19 至向量 22 的中断因素。本寄存器在 IRQQCMODE:IRQCMODE 位为 "1" 时有效。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	INTSEL3								INTSEL2							
属性	R/W								R/W							
初始值	0x00								0x00							

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	INTSEL1								INTSEL0							
属性	R/W								R/W							
初始值	0x00								0x00							

寄存器功能

[bit31:24] INTSEL3:
这些位选择*中断向量 22 的中断因素。

[bit23:16] INTSEL2:
这些位选择*中断向量 21 的中断因素。

[bit15:8] INTSEL1:
这些位选择*中断向量 20 的中断因素。

[bit7:0] INTSEL0:
这些位选择*中断向量 19 的中断因素。

*: 有关中断因素的选择，详见 Table 2-3。

注意事项:

- *RCINTSEL0 选择的中断因素受 IRQ11~IRQ31 屏蔽。(IRQ11MON ~ IRQ31MON 寄存器中的可用位也被屏蔽。)*
- *选择中断因素时，确认从 INTSEL0 ~ INTSEL7 位选择的所有中断因素各不相同。*

2.23 中断因素选择寄存器 1 (RCINTSEL1)

中断因素选择寄存器 1 (RCINTSEL1) 选择中断向量 23 至向量 26 的中断因素。本寄存器在 IRQQCMODE:IRQCMODE 位为 "1" 时有效。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	INTSEL7								INTSEL6							
属性	R/W								R/W							
初始值	0x00								0x00							

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	INTSEL5								INTSEL4							
属性	R/W								R/W							
初始值	0x00								0x00							

寄存器功能

[bit31:24] INTSEL7:
这些位选择*中断向量 26 的中断因素。

[bit23:16] INTSEL6:
这些位选择*中断向量 25 的中断因素。

[bit15:8] INTSEL5:
这些位选择*中断向量 24 的中断因素。

[bit7:0] INTSEL4:
这些位选择*中断向量 23 的中断因素。

*: 有关中断因素的选择，详见 Table 2-3。

注意事项:

- *RCINTSEL0 选择的中断因素受 IRQ11~IRQ31 屏蔽。(IRQ11MON ~ IRQ31MON 寄存器中的可用位也被屏蔽。)*
- *选择中断因素时，确认从 INTSEL0 ~ INTSEL7 位选择的所有中断因素各不相同。*

Table 2-3 中断因素选择

设置 RCINTSELx:INTSELx	中断因素
0x00	未选择中断因素。
0x01	外部中断 ch.0
0x02	外部中断 ch.1
0x03	外部中断 ch.2
0x04	外部中断 ch.3
0x05	外部中断 ch.4
0x06	外部中断 ch.5
0x07	外部中断 ch.6
0x08	外部中断 ch.7
0x09	外部中断 ch.8
0x0A	外部中断 ch.9
0x0B	外部中断 ch.10
0x0C	外部中断 ch.11
0x0D	基本计时器 ch.0 的 IRQ0/IRQ1
0x0E	基本计时器 ch.1 的 IRQ0/IRQ1
0x0F	基本计时器 ch.2 的 IRQ0/IRQ1
0x10	基本计时器 ch.3 的 IRQ0/IRQ1
0x11	基本计时器 ch.4 的 IRQ0/IRQ1
0x12	基本计时器 ch.5 的 IRQ0/IRQ1
0x13	基本计时器 ch.6 的 IRQ0/IRQ1
0x14	基本计时器 ch.7 的 IRQ0/IRQ1
0x15	MFS ch.0 的接收中断
0x16	MFS ch.1 的接收中断
0x17	MFS ch.2 的接收中断
0x18	MFS ch.3 的接收中断
0x19	MFT 单元 0 自由运行计时器 ch.0 的零检测中断
0x1A	MFT 单元 1 自由运行计时器 ch.0 的零检测中断
0x1B	MFT 单元 2 自由运行计时器 ch.0 的零检测中断
0x1C	DMAC ch.0
0x1D	DMAC ch.1
0x1E	DMAC ch.2
0x1F	DMAC ch.3
0x20	MFS ch.8 的接收中断
0x21	MFS ch.9 的接收中断
0x22	MFS ch.10 的接收中断
0x23	MFS ch.11 的接收中断
0x24	MFS ch.12 的接收中断

设置 RCINTSELx:INTSELx	中断因素
0x25	MFS ch.13 的接收中断
0x26	MFS ch.14 的接收中断
0x27	MFS ch.15 的接收中断
0x28	MFS ch.8 的传输/状态中断
0x29	MFS ch.9 的传输/状态中断
0x2A	MFS ch.10 的传输/状态中断
0x2B	MFS ch.11 的传输/状态中断
0x2C	MFS ch.12 的传输/状态中断
0x2D	MFS ch.13 的传输/状态中断
0x2E	MFS ch.14 的传输/状态中断
0x2F	MFS ch.15 的传输/状态中断
0x30 to 0xFF	保留

3. 使用注意事项

使用中断控制器时应注意以下各点：

- 各外设资源的中断请求信号按级别通知。从中断处理退出时，总是清除中断请求。
- 分配 NMIX 引脚复用为通用端口。重置释放后的初始值作为通用端口配置，且 NMI 输入被屏蔽。使用 NMI 功能时，需要配置端口设置以使能 NMI 功能。详见“外部中断和 NMI 控制单元”一章。
- 各外设资源的具体事件检测寄存器与中断使能寄存器的相互关系，参见各模块相关章节。

第 7-5 章：中断(TYPE 2)

本章说明中断控制器（**TYPE　2**）和外设中断请求（**TYPE　2**）。

1. **概述**
2. **配置**

1. 概述

Cortex- M0 + CPU 内核配备了嵌套向量中断控制器(NVIC)。NVIC 支持预留系统异常和 32 个外设中断，还可设置 4 个中断优先级的优先顺序（通过内置 2 位寄存器）。本节说明微控制器内安装外设功能的中断信号（TYPE 2）以及 NVIC 和中断信号（TYPE 2）之间的连接。

2. 配置

框图

Figure 1-1 中断信号（TYPE 2）与 NVIC 间的连接

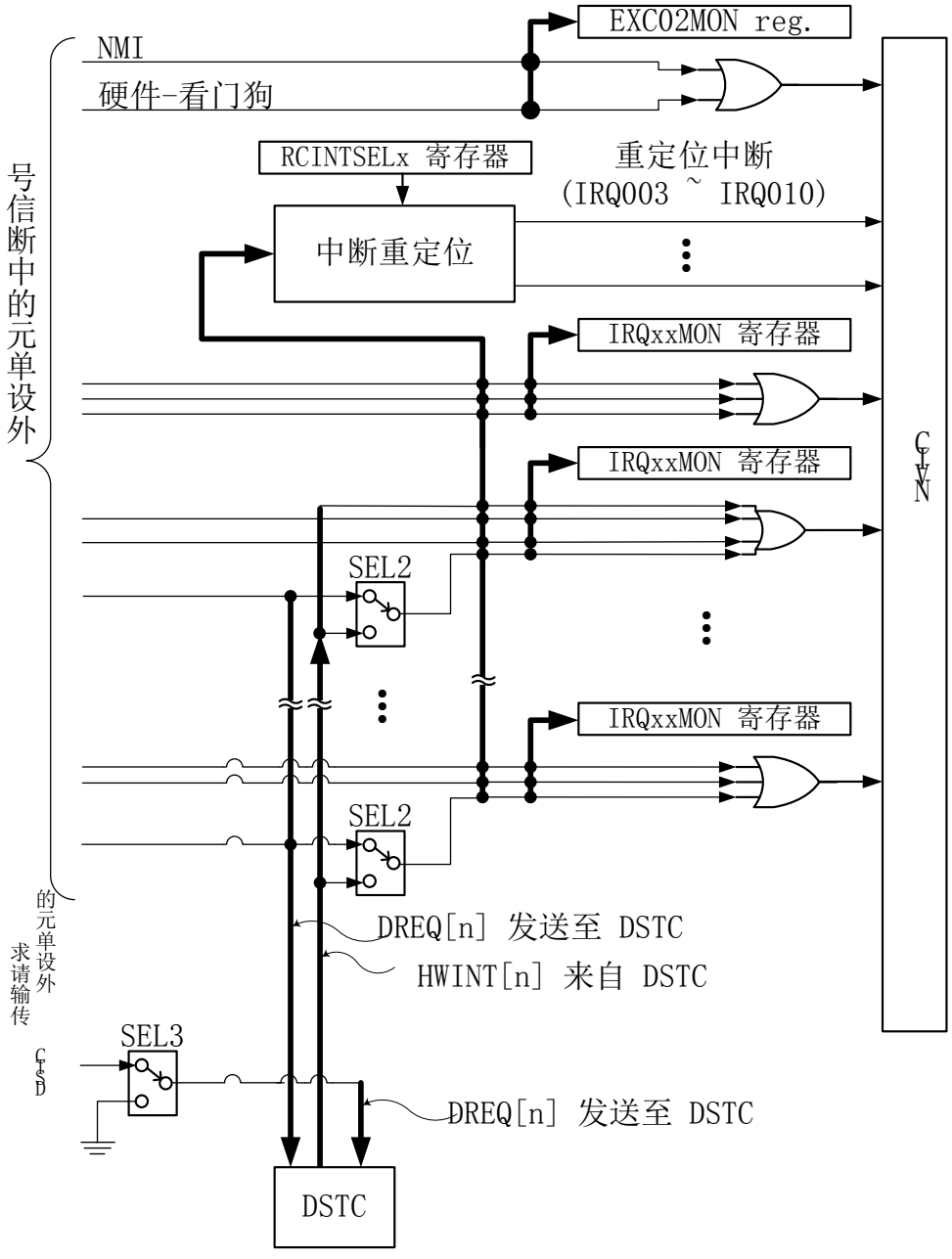


Figure 1-1 说明了 NVIC 如何连接到外设功能输入中断信号，以及 DSTC。详细连接说明如下：

NVIC

NVIC 支持预留系统异常和 32 个外设中断。有关 NVIC 的详细内容，参见《Cortex-M0+ 技术参考手册》。《Cortex-M0+ 技术参考手册》中，将预留系统异常之外的异常定义为 “外部中断 (IRQ)。”为了区别于发自微控制器外部输入引脚的外部中断 (IRQ) ，本文档中外部中断 (IRQ) 称为外设中断。

NVIC 的中断优先寄存器为 2 位配置，可设置 4 个中断优先级。

可使用 NVIC 安装的系统处理程序优先级寄存器（地址: 0xE000ED18、0xE000ED1C、0xE000ED20）设置预留系统异常 4 ~ 15 的各优先级。

可使用 NVIC 内安装的 IRQ 优先级寄存器（地址： 0xE000E400 ~ 0xE000E41C）设置外设中断异常 16 ~ 47 的优先级。

NVIC 支持不可屏蔽中断 (NMI) 输入。

中断集合和批量读取寄存器

从各外设功能输入的中断信号(Figure 1-1 中外设单元的中断信号)由图中所示逻辑 “或” 电路集合。然后将集合的中断信号连接至 NVIC 的 32 个外设中断的其中一个。参见 Table 4-1，检查 NVIC 的中断输出分配至哪一个外设功能。

由于中断信号通过逻辑 “或” 电路集合，因此多个源生成一个 NVIC 中断。生成中断后，通过读取中断批量读取寄存器（Figure 1-1 所示 IRQxxxMON 寄存器）识别该中断的生成源。中断批量读取寄存器 (IRQ000MON ~ IRQ31MON) 包括 NVIC 的所有中断输入。

若产品未配有此功能，则 IRQxxxMON 寄存器的各位均为保留位。

外部中断和 NMI 控制器的不可屏蔽中断信号 (NMI)，及硬件看门狗计时器的中断信号（HW-看门狗）通过逻辑 “或” 电路集合，然后连接至 NVIC 异常 2 的输入。生成异常 2 的中断后，中断源（要么是外部中断和 NMI 控制器，要么硬件看门狗计时器）可通过读取 EXC02MON 寄存器识别。

微控制器的 NMI 引脚复用为通用端口。重置释放后，引脚的初始功能为通用端口，且 NMI 输入被屏蔽。使用 NMI 功能时，需要通过 I/O 端口设置使能 NMI 功能。详见 “I/O 端口” 一章。通过外部中断和 NMI 控制器将 NMI 输入信号输入 NVIC。

中断重定位功能

可通过 IRQCMODE 寄存器设置选择 Table 2 所示的两类中断因素向量。有关 IRQCMODE 位,参见 “中断 (B)”一章。有关设置详细内容，参见 Table 1-1 所列各章。
此外，可通过 RCINTSEL0 和 RCINTSEL1 寄存器选择任意中断因素。有关 RCINTSEL0 和 RCINTSEL1 寄存器，详见 "中断 (B)" 一章。

Table 1-1 中断章节对应表

IRQCMODE 设置	参考章节
IRQCMODE=0 未选择重定位	"中断 (A)"章节
IRQCMODE=1 选择重定位	"中断 (B)"章节

DSTC 传输请求和 DSTC 传输结束通知选择

有两类外设功能使用 DSTC 的 DMA 传输：一类将中断用为向 DSTC 发出的传输请求，另一类分别处理为中断和向 DSTC 发出的传输请求。

如果外设功能将中断用为 DSTC 的传输请求，则此外设功能可将中断信号用作对 DSTC 的 DMA 传输请求信号。通过 DSTC 的 DREQENB[n] 寄存器设置，将外设功能的中断信号识别为 DMA 传输请求。

如果外设功能分别处理中断和对 DSTC 的传输请求，则此外设功能为可编程 CRC。此外设功能分别保持来自中断的 DSTC 传输请求。

Table 1-1 列出各功能之间的差异。

Table 1-2 使用 DSTC 的外设功能差异

1. 外设功能类型	2. DREQENB[n]=0 时			3. DREQENB[n]=1 时		
	电路状态	IRQxxxMON	通知 NVIC	电路状态	IRQxxxMON	通知 NVIC
外设功能分别处理中断和对 DSTC 的传输请求（可编程 CRC）	SEL3 选择 GND	显示外设功能的中断。（未发生 DSTC 的传输完成中断）	外设功能的中断（未发生 DSTC 的传输完成中断）	SEL3 选择外设功能的 DMA 传输请求。	外设功能的中断和 DSTC 的传输完成中断分别由不同寄存器显示。	外设功能的中断和 DSTC 的中断
外设功能（除可编程 CRC 外的使用 DSTC 传输的外设功能）将中断用作对 DSTC 的传输请求	SEL2 选择外设功能的中断	显示外设功能的中断。	外设功能的中断	SEL2 选择 DSTC 的中断。	显示 DSTC 的中断。	DSTC 的中断

若外设功能将中断用作对 DSTC 的传输请求，选择器电路（图中所示 SEL2）或选择将 DSTC 的 DMA 传输完成通知信号输出（图中所示 DSTC 的 HWINT[n]），或选择外设功能的中断信号作为中断信号输入至 NVIC。 SEL2 的切换根据 DREQENB[n] 寄存器设置执行。

若选择 DSTC 的 DMA 传输，则发生 DSTC 的传输完成中断，而不是外设的中断。由于图中所示电路配置原因，可从 IRQxxMON 寄存器的对应位读取 DSTC 的传输完成中断。此外，可应用中断重定位功能。

若外设功能分别处理为中断和对 DSTC 发出的传输请求，则通过 DREQENB 寄存器设置确定是否将外设功能的 DSTC 传输请求连接至 DSTC（图中所示 SEL3）。此外，这种情况下外设功能的中断以及 DSTC 的传输完成中断分别输入至 NVIC，无需选择器电路（图中所示 SEL2）。可从不同 IRQxxxMON 寄存器的对应位分别读取 DSTC 的传输完成中断及外设功能的中断。此外，可应用中断重定位功能。

有关各外设功能的 DSTC 传输请求，参见各外设功能章节。

DMA 传输接受信号连接

有些外设功能模块中，向 DMA 传输至这些外设结束后必须清除传输请求信号（中断信号）。这类外设功能的传输请求信号通过 DSTC 清除。若在 SEL2 中选择由 DSTC 进行 DMA 传输，DSTC 的 DMA 传输确认信号（FIGURE 1-1 中未说明）连接至外设功能。

DMA 传输停止信号连接

DMA 传输停止请求信号从多功能串行单元（下文称为 "MFS"）输出。根据 SEL2 中的选择， MFS（FIGURE 1-1 未说明）连接至 DSTC，具体说明如下。

如果在 SEL2 中如选择 DSTC 与 MFS 间的连接，DSTC 根据传输停止请求信号停止传输操作 MFS 的传输停止请求信号取反之前，DSTC 不能执行传输操作。MFS 的传输停止请求信号由逻辑“或”通过 DSTC 的传输结束中断（HWINT[n] 信号）集合并作为中断信号通知 NVIC。

第 7-6 章：中断 (TYPE2-A)

本章说明 **IRQCMODE=0** 时的异常、中断因素向量及寄存器。

1. 中断列表
2. 寄存器
3. 使用注意事项

代码：9BFIRQC_B_FM0+-C01.0

1. 中断列表

本节列出输入至 NVIC 的异常和中断源，以及由 DSTC 的 DMA 传输所传输的中断。

1.1 异常及中断列表

Table 1-1 列出待输入至 NVIC 的异常及中断源。表中各列详细内容如下：

Exc 号
NVIC 异常编号

IRQ 号
外设中断编号（编号=Exc 号 - 16）

向量偏移
中断引用的向量存储偏移地址

位
指批量读取寄存器（IRQxxxMON 或 EXC02MON）中数位的编号，通过此位读取中断源。若单个 IRQ 有多个位编号，则多个源通过逻辑“或”集合，并从其对应位读出中断源。若单个 IRQ 仅有位编号 "0"，则没有逻辑“或”聚合的多个源。本列中 "-" 表示没有该异常或中断的批量读取寄存器。

DSTC
本列中的数字，按描述值表示将中断用为向 DSTC 发出传输请求的外设功能，也是与 DSTC 的 DMA 传输兼容的中断信号。此编号表示 DSTC 的 DREQENB[n] 寄存器的位编号。在此情况下，DSTC 的 DREQENB[n] 寄存器设置确定 Figure 1-1 中 SEL2 的连接。

"*" 按描述值表示这是分别处理中断和对 DSTC 发出传输请求的外设功能，并且只输入通过 DSTC 进行的 DMA 传输完成中断。外设功能的传输请求信号不输入至 NVIC ，而传输完成后从 DSTC 生成的传输完成中断 (HWINT[n]) 输入至 NVIC，无需 Figure 1-1 所示选择器电路 (SEL2)。

"-" 按描述值表示与 DSTC 的 DMA 传输不兼容。

异常源和中断源
本列包括异常源和中断源。某些中断有多个源。外设功能的此类中断源通过逻辑“或”集合。即使仅显示外设源的一个中断源，这类外设功能仍可能有逻辑“或”集合的多个中断源。详见各外设功能相关说明。

Table 1-1 异常源及中断源 (IRQCMODE=0)列表

Exc. 号	IRQ 号	向量偏移	位	DSTC	异常及中断源
0	-	0x000	-	-	（栈指针初始值）
1	-	0x004	-	-	复位
2	-	0x008	1	-	硬件看门狗计时器中断
			0	-	不可屏蔽中断 (NMI)
3	-	0x00C	-	-	硬故障
4	-	0x010	-	-	保留
5	-	0x014	-	-	保留
6	-	0x018	-	-	保留
7 ~ 10	-	0x01C | 0x02B	-	-	保留
11	-	0x02C	-	-	SVCall（监视器调用）
12	-	0x030	-	-	保留
13	-	0x034	-	-	保留
14	-	0x038	-	-	PendSV
15	-	0x03C	-	-	SysTick
16	0	0x040	0	-	时钟监视器的异常频率检测中断 (FCS)
17	1	0x044	0	-	软件看门狗计时器中断
18	2	0x048	0	-	低压检测 (LVD) 中断
19	3	0x04C	3	25	MFT 单元 0 WFG 计时器 54 中断
			2	24	MFT 单元 0 WFG 计时器 32 中断
			1	23	MFT 单元 0 WFG 计时器 10 中断
			0	-	MFT 单元 0 DTIF（电机紧急停止）中断
20	4	0x050	7	7	外部引脚中断 ch.7
			6	6	外部引脚中断 ch.6
			5	5	外部引脚中断 ch.5
			4	4	外部引脚中断 ch.4
			3	3	外部引脚中断 ch.3
			2	2	外部引脚中断 ch.2
			1	1	外部引脚中断 ch.1
			0	0	外部引脚中断 ch.0

Exc. 号	IRQ 号	向量偏移	位	DSTC	异常及中断源
21	5	0x054	15	-	外部引脚中断 ch.23
			14	-	外部引脚中断 ch.22
			13	-	外部引脚中断 ch.21
			12	-	外部引脚中断 ch.20
			11	-	外部引脚中断 ch.19
			10	-	外部引脚中断 ch.18
			9	-	外部引脚中断 ch.17
			8	-	外部引脚中断 ch.16
			7	-	外部引脚中断 ch.15
			6	-	外部引脚中断 ch.14
			5	-	外部引脚中断 ch.13
			4	12	外部引脚中断 ch.12
			3	11	外部引脚中断 ch.11
			2	10	外部引脚中断 ch.10
			1	9	外部引脚中断 ch.9
			0	8	外部引脚中断 ch.8
22	6	0x058	1	-	双计时器中断 ch.2
			0	-	双计时器中断 ch.1
23	7	0x05C	0	46	MFS ch.0 的接收中断
24	8	0x060	1	-	MFS ch.0 状态中断
			0	47	MFS ch.0 传输中断
25	9	0x064	0	48	MFS ch.1 接收中断
26	10	0x068	1	-	MFS ch.1 状态中断
			0	49	MFS ch.1 传输中断
27	11	0x06C	0	50	MFS ch.2 接收中断
28	12	0x070	1	-	MFS ch.2 状态中断
			0	51	MFS ch.2 传输中断
29	13	0x074	0	52	MFS ch.3 接收中断
30	14	0x078	1	-	MFS ch.3 状态中断
			0	53	MFS ch.3 传输中断
31	15	0x07C	0	54	MFS ch.4 接收中断
32	16	0x080	1	-	MFS ch.4 状态中断
			0	55	MFS ch.4 传输中断
33	17	0x084	0	56	MFS ch.5 接收中断
34	18	0x088	1	-	MFS ch.5 状态中断
			0	57	MFS ch.5 传输中断
35	19	0x08C	0	58	MFS ch.6 接收中断
36	20	0x090	1	-	MFS ch.6 状态中断
			0	59	MFS ch.6 传输中断
37	21	0x094	0	60	MFS ch.7 接收中断
38	22	0x098	1	-	MFS ch.7 状态中断
			0	61	MFS ch.7 传输中断

Exc. 号	IRQ 号	向量偏移	位	DSTC	异常及中断源
39	23	0x09C	12	-	DSTC ERINT 中断
			11	-	DSTC SWINT 中断
			10	-	智能卡（IC 卡）中断 ch.0
			9	-	HDMI-CEC 遥控接收中断 ch.0
			8	-	保留
			7	-	保留
			6	-	保留
			5	-	保留
			4	-	保留
			3	-	保留
			2	44	PPG 中断 ch.4
			1	43	PPG 中断 ch.2
			0	42	PPG 中断 ch.0
40	24	0x0A0	10	-	智能卡（IC 卡）中断 ch.1
			9	-	HDMI-CEC 遥控接收中断 ch.1
			8~6	-	保留
			5	-	实时时钟计数器中断
			4	45	计时计数器中断
			3	-	USB PLL 振荡稳定等待完成中断
			2	-	主 PLL 振荡稳定等待完成中断
			1	-	副时钟振荡稳定等待完成中断
			0	-	主时钟振荡稳定等待完成中断
41	25	0x0A4	4	-	ADC 单元 0 范围比较结果中断
			3	-	ADC 单元 0 结果比较中断
			2	-	ADC 单元 0 FIFO 溢出中断
			1	62	ADC 单元 0 扫描转换中断
			0	63	ADC 单元 0 优先转换中断
42	26	0x0A8	5	-	USB ch.0 主机 SOFIRQ 中断 USB ch.0 主机 CMPIRQ 中断
			4	-	USB ch.0 主机 DIRQ 中断 USB ch.0 主机 URIRQ 中断 USB ch.0 主机 RWKIRQ 中断 USB ch.0 主机 CNNIRQ 中断
			3	-	USB ch.0 从机 SPK 中断
			2	-	USB ch.0 从机 SUSP 中断 USB ch.0 从机 SOF 中断 USB ch.0 从机 BRST 中断 USB ch.0 从机 CONF 中断 USB ch.0 从机 WKUP 中断
			1	-	USB ch.0 从机端点 0 DRQO 中断
			0	-	USB ch.0 从机端点 0 DRQI 中断
43	27	0x0AC	5	-	LCD 控制器中断
			4~0	-	保留

Exc. 号	IRQ 号	向量偏移	位	DSTC	异常及中断源
44	28	0x0B0	5	31	MFT 单元 0 自由运行计时器零检测中断 ch.2
			4	30	MFT 单元 0 自由运行计时器零检测中断 ch.1
			3	29	MFT 单元 0 自由运行计时器零检测中断 ch.0
			2	28	MFT 单元 0 自由运行计时器峰值检测中断 ch.2
			1	27	MFT 单元 0 自由运行计时器峰值检测中断 ch.1
			0	26	MFT 单元 0 自由运行计时器峰值检测中断 ch.0
45	29	0x0B4	8	-	USB ch.0 从机端点 5 DRQ 中断
			7	-	USB ch.0 从机端点 4 DRQ 中断
			6	-	USB ch.0 从机端点 3 DRQ 中断
			5	-	USB ch.0 从机端点 2 DRQ 中断
			4	-	USB ch.0 从机端点 1 DRQ 中断
			3	35	MFT 单元 0 ICU 输入沿检测中断 ch.3
			2	34	MFT 单元 0 ICU 输入沿检测中断 ch.2
			1	33	MFT 单元 0 ICU 输入沿检测中断 ch.1
			0	32	MFT 单元 0 ICU 输入沿检测中断 ch.0
46	30	0x0B8	5	41	MFT 单元 0 OCU 匹配检测中断 ch.5
			4	40	MFT 单元 0 OCU 匹配检测中断 ch.4
			3	39	MFT 单元 0 OCU 匹配检测中断 ch.3
			2	38	MFT 单元 0 OCU 匹配检测中断 ch.2
			1	37	MFT 单元 0 OCU 匹配检测中断 ch.1
			0	36	MFT 单元 0 OCU 匹配检测中断 ch.0
47	31	0x0BC	27	-	闪存 RDY、HANG 中断
			26~16	-	保留
			15	-	基本计时器 ch.7 源 1 (IRQ1)
			14	-	基本计时器 ch.7 源 0 (IRQ0)
			13	-	基本计时器 ch.6 源 1 (IRQ1)
			12	22	基本计时器 ch.6 源 0 (IRQ0)
			11	-	基本计时器 ch.5 源 1 (IRQ1)
			10	-	基本计时器 ch.5 源 0 (IRQ0)
			9	-	基本计时器 ch.4 源 1 (IRQ1)
			8	21	基本计时器 ch.4 源 0 (IRQ0)
			7	20	基本计时器 ch.3 源 1 (IRQ1)
			6	19	基本计时器 ch.3 源 0 (IRQ0)
			5	18	基本计时器 ch.2 源 1 (IRQ1)
			4	17	基本计时器 ch.2 源 0 (IRQ0)
			3	16	基本计时器 ch.1 源 1 (IRQ1)
			2	15	基本计时器 ch.1 源 0 (IRQ0)
			1	14	基本计时器 ch.0 源 1 (IRQ1)
			0	13	基本计时器 ch.0 源 0 (IRQ0)

1.2 输入至 DSTC 的中断信号

Table 1-2 列出作为传输请求信号输入至 DSTC 的中断信号。表中编号对应 DSTC 的 DREQENB[n] 寄存器编号。

Table 1-2 输入至 DSTC 的中断信号列表

编号	中断信号名称
0	外部引脚中断 ch.0
1	外部引脚中断 ch.1
2	外部引脚中断 ch.2
3	外部引脚中断 ch.3
4	外部引脚中断 ch.4
5	外部引脚中断 ch.5
6	外部引脚中断 ch.6
7	外部引脚中断 ch.7
8	外部引脚中断 ch.8
9	外部引脚中断 ch.9
10	外部引脚中断 ch.10
11	外部引脚中断 ch.11
12	外部引脚中断 ch.12
13	基本定时器 ch.0 源 0 (IRQ0) 中断
14	基本定时器 ch.0 源 1 (IRQ1) 中断
15	基本定时器 ch.1 源 0 (IRQ0) 中断
16	基本定时器 ch.1 源 1 (IRQ1) 中断
17	基本定时器 ch.2 源 0 (IRQ0) 中断
18	基本定时器 ch.2 源 1 (IRQ1) 中断
19	基本定时器 ch.3 源 0 (IRQ0) 中断
20	基本定时器 ch.3 源 1 (IRQ1) 中断
21	基本定时器 ch.4 源 0 (IRQ0) 中断
22	基本定时器 ch.6 源 0 (IRQ0) 中断
23	MFT 单元 0 WFG 计时器 10 中断
24	MFT 单元 0 WFG 计时器 32 中断
25	MFT 单元 0 WFG 计时器 54 中断
26	MFT 单元 0 FRT ch.0 峰值检测中断
27	MFT 单元 0 FRT ch.1 峰值检测中断
28	MFT 单元 0 FRT ch.2 峰值检测中断
29	MFT 单元 0 FRT ch.0 零检测中断
30	MFT 单元 0 FRT ch.1 零检测中断
31	MFT 单元 0 FRT ch.2 零检测中断
32	MFT 单元 0 ICU ch.0 输入沿检测中断
33	MFT 单元 0 ICU ch.1 输入沿检测中断
34	MFT 单元 0 ICU ch.2 输入沿检测中断
35	MFT 单元 0 ICU ch.3 输入沿检测中断
36	MFT 单元 0 OCU ch.0 匹配检测中断
37	MFT 单元 0 OCU ch.1 匹配检测中断
38	MFT 单元 0 OCU ch.2 匹配检测中断

编号	中断信号名称
39	MFT 单元 0 OCU ch.3 匹配检测中断
40	MFT 单元 0 OCU ch.4 匹配检测中断
41	MFT 单元 0 OCU ch.5 匹配检测中断
42	PPG ch.0 中断
43	PPG ch.2 中断
44	PPG ch.4 中断
45	计时计数器中断
46	MFS ch.0 接收中断
47	MFS ch.0 传输中断
48	MFS ch.1 接收中断
49	MFS ch.1 传输中断
50	MFS ch.2 接收中断
51	MFS ch.2 传输中断
52	MFS ch.3 接收中断
53	MFS ch.3 传输中断
54	MFS ch.4 接收中断
55	MFS ch.4 传输中断
56	MFS ch.5 接收中断
57	MFS ch.5 传输中断
58	MFS ch.6 接收中断
59	MFS ch.6 传输中断
60	MFS ch.7 接收中断
61	MFS ch.7 传输中断
62	A/D 转换器单元 0 扫描转换中断
63	A/D 转换器单元 0 优先转换中断

2. 寄存器

本节说明各寄存器详情。

寄存器表

缩写	寄存器名称	参考章节
EXC02MON	EXC02 批量读取寄存器	2.1
IRQ00MON	IRQ000 批量读取寄存器	2.2
IRQ01MON	IRQ001 批量读取寄存器	
IRQ02MON	IRQ002 批量读取寄存器	
IRQ03MON	IRQ003 批量读取寄存器	
IRQ04MON	IRQ004 批量读取寄存器	
IRQ05MON	IRQ005 批量读取寄存器	
IRQ06MON	IRQ006 批量读取寄存器	
IRQ07MON	IRQ007 批量读取寄存器	
IRQ08MON	IRQ008 批量读取寄存器	
IRQ09MON	IRQ009 批量读取寄存器	
IRQ10MON	IRQ010 批量读取寄存器	
IRQ11MON	IRQ011 批量读取寄存器	
IRQ12MON	IRQ012 批量读取寄存器	
IRQ13MON	IRQ013 批量读取寄存器	
IRQ14MON	IRQ014 批量读取寄存器	
IRQ15MON	IRQ015 批量读取寄存器	
IRQ16MON	IRQ016 批量读取寄存器	
IRQ17MON	IRQ017 批量读取寄存器	
IRQ18MON	IRQ018 批量读取寄存器	
IRQ19MON	IRQ019 批量读取寄存器	
IRQ20MON	IRQ020 批量读取寄存器	
IRQ21MON	IRQ021 批量读取寄存器	
IRQ22MON	IRQ022 批量读取寄存器	
IRQ23MON	IRQ023 批量读取寄存器	
IRQ24MON	IRQ024 批量读取寄存器	
IRQ25MON	IRQ025 批量读取寄存器	
IRQ26MON	IRQ026 批量读取寄存器	
IRQ27MON	IRQ027 批量读取寄存器	
IRQ28MON	IRQ028 批量读取寄存器	
IRQ29MON	IRQ029 批量读取寄存器	
IRQ30MON	IRQ030 批量读取寄存器	
IRQ31MON	IRQ031 批量读取寄存器	

2.1 EXC02 批量读取寄存器 (EXC02MON)

EXC02 批量读取寄存器 (EXC02MON) 可一次读取分配给异常 2 的所有中断（NMI 和硬件看门狗中断）。

寄存器配置

位	31							8
字段	保留							
属性	R							
初始值	0x000000							

位	7	6	5	4	3	2	1	0
字段	保留						HWINT	NMI
属性	R						R	R
初始值	000000						0	0

寄存器功能

[bit31:2] 保留：保留位

保留位读为 “0”。

[bit1] HWINT

值	描述
0	没有硬件看门狗计时器的中断请求。
1	已发出硬件看门狗计时器的中断请求。

[bit0] NMI

值	描述
0	没有 NMIX 外部引脚的中断请求。
1	已发出 NMIX 外部引脚的中断请求。

2.2 IRQxx 批量读取寄存器 (IRQxxMON)

IRQxx 批量读取寄存器表示 32 个寄存器（即：IRQ00MON – IRQ31MON ）。它们分别与 NVIC 的 IRQ00 – IRQ31 中断输入对应。逻辑“或”电路集合的中断信号状态可通过这些寄存器可读出。

参见 Table 1-1，检查此寄存器的哪一位分配给外设功能的哪一个中断。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	INT31	INT30	INT29	INT28	INT27	INT26	INT25	INT24
属性	R	R	R	R	R	R	R	R
初始值	0	0	0	0	0	0	0	0

位	23	22	21	20	19	18	17	16
字段	INT23	INT22	INT21	INT20	INT19	INT18	INT17	INT16
属性	R	R	R	R	R	R	R	R
初始值	0	0	0	0	0	0	0	0

位	15	14	13	12	11	10	9	8
字段	INT15	INT14	INT13	INT12	INT11	INT10	INT09	INT08
属性	R	R	R	R	R	R	R	R
初始值	0	0	0	0	0	0	0	

位	7	6	5	4	3	2	1	0
字段	INT07	INT06	INT05	INT04	INT03	INT02	INT01	INT00
属性	R	R	R	R	R	R	R	R
初始值	0	0	0	0	0	0	0	0

寄存器功能

[bit31:0] INT31 – INT00

操作	描述
写入	忽略写访问。
读取 0	分配给此位的中断未发生。
读取 1	发生分配给此位的中断。

3. 使用注意事项

使用中断控制器时应注意以下各点：

- 按级别将外设功能的中断请求信号通知中断控制器。退出中断处理时，总是清除该中断的中断请求。
- 进程中有其它优先级更高的中断请求时，通知 NVIC 的外设功能中断请求会在 NVIC 内挂起。取消 NVIC 中挂起的中断请求时，清除外设功能的中断请求，并通过 NVIC 内安装的中断清除挂起寄存器（地址：0xE000E280 ~ 0xE000E29C）清除 NVIC 中挂起的中断请求。
- NMIX 引脚复用为通用端口。重置释放后，引脚的初始功能为通用端口，且 NMI 中断被屏蔽。使用 NMI 功能时，需通过端口设置使能 NMI 功能。详见“ 外部中断和 NMI 控制器”一章。
- 若使用由 DSTC 进行的 DMA 传输，将生成 DSTC 的传输结束中断 (HWINT[n])，而不是外设功能的中断。由于以上配置的原因，NVIC 产生外设功能的中断，且 DSTC 的传输结束中断跳转至同一中断向量。使用 DREQENB[n] 寄存器选择待处理的中断。
  但对于有些分别处理中断和向 DSTC 发出传输请求的外设功能（MFSI2S、HS-SPICNT、CAN-FD 及可编程 CRC），DSTC 的 DREQENB 寄存器设置确定是否执行 DMA 传输。在此情况下，外设功能的中断和 DSTC 的传输完成中断分别输入至 NVIC。
- 有关外设资源内具体事件检测寄存器和中断使能寄存器的关系，参见外设功能一章。

第 7-7 章：中断 (TYPE 2-B)

本章说明 **IRQCMODE=1** 时的异常、中断因素向量及寄存器。

1. 中断列表
2. 寄存器
3. 使用注意事项

代码：9BFIRQC_B_FM0-C01.0

1. 中断列表

本节列出输入至 NVIC 的异常和中断源，以及由 DSTC 传输的 DMA 传输中断。

1.1 异常及中断列表

Table 1-1 列出待输入至 NVIC 的异常及中断源。表中各列详细内容如下：

Exc 号
NVIC 异常编号

IRQ 号
外设中断编号（编号=Exc 号 - 16）

向量偏移
中断引用的向量存储偏移地址

位
指批量读取寄存器（IRQxxxMON 或 EXC02MON）中数位的编号，通过此位读取中断源。若单个 IRQ 有多个位编号，则多个源通过逻辑“或”集合，并从其对应位读出中断源。若单个 IRQ 仅有位编号 "0"，则没有逻辑“或”集合的多个源。本列中 "-" 表示没有该异常或中断的批量读取寄存器。

DSTC
本列中的数字编号表示将中断作为传输请求发送至 DSTC 的外设功能，也是与 DSTC 的 DMA 传输兼容的中断信号。此编号表示 DSTC 的 DREQENB[n] 寄存器的位编号。在此情况下，DSTC 的 DREQENB[n] 寄存器设置确定 figure 1-1 中 SEL2 的连接。

"*" 表示这是分别处理中断和对 DSTC 发出传输请求的外设功能，并且只输入通过 DSTC 进行的 DMA 传输完成中断。外设功能的传输请求信号不输入至 NVIC ，而传输完成后从 DSTC 生成的传输完成中断 (HWINT[n]) 输入至 NVIC，无需 Figure 1-1 所示选择器电路 (SEL2)。

"-" 表示与 DSTC 的 DMA 传输不兼容。

异常源和中断源
本列包括异常源和中断源。某些中断有多个源。外设功能的此类中断源通过逻辑“或”集合。即使仅显示外设源的一个中断源，这类外设功能仍可能有逻辑 “或”集合的多个中断源。详见各外设功能相关说明。

Table 1-1 异常源及中断源 (IRQCMODE=1)列表

Exc. 号	IRQ 号	向量偏移	位	DSTC	异常和中断源
0	-	0x000	-	-	（栈指针初始值）
1	-	0x004	-	-	复位
2	-	0x008	1	-	硬件看门狗计时器中断
			0	-	不可屏蔽中断 (NMI)
3	-	0x00C	-	-	硬件故障
4	-	0x010	-	-	保留
5	-	0x014	-	-	保留
6	-	0x018	-	-	保留
7 ~ 10	-	0x01C | 0x02B	-	-	保留
11	-	0x02C	-	-	SVCall（监视器调用）
12	-	0x030	-	-	保留
13	-	0x034	-	-	保留
14	-	0x038	-	-	PendSV
15	-	0x03C	-	-	SysTick
16	0	0x040	0	-	时钟监视器的异常频率检测中断 (FCS)
17	1	0x044	0	-	软件看门狗计时器中断
18	2	0x048	0	-	低压检测 (LVD) 中断
19	3	0x04C	0	-	可重定位中断 （由 INTSEL0:RCINTSEL0 寄存器选择）
20	4	0x050	0	-	可重定位中断 （由 INTSEL1:RCINTSEL0 寄存器选择）
21	5	0x054	0	-	可重定位中断 （由 INTSEL2:RCINTSEL0 寄存器选择）
22	6	0x058	0	-	可重定位中断 （由 INTSEL3:RCINTSEL0 寄存器选择）
23	7	0x05C	0	-	可重定位中断 （由 INTSEL4:RCINTSEL1 寄存器选择）
24	8	0x060	0	-	可重定位中断 （由 INTSEL5:RCINTSEL1 寄存器选择）
25	9	0x064	0	-	可重定位中断 （由 INTSEL6:RCINTSEL1 寄存器选择）
26	10	0x068	0	-	可重定位中断 （由 INTSEL7:RCINTSEL1 寄存器选择）
27	11	0x06C	3	25	MFT 单元 0 波形发生器 54
			2	24	MFT 单元 0 波形发生器 32
			1	23	MFT 单元 0 波形发生器 10
			0	-	MFT 单元 0 DTIF（电机紧急停止）中断

Exc. 号	IRQ 号	向量偏移	位	DSTC	异常和中断源
28	12	0x070	7	7	外部引脚中断 ch.7
			6	6	外部引脚中断 ch.6
			5	5	外部引脚中断 ch.5
			4	4	外部引脚中断 ch.4
			3	3	外部引脚中断 ch.3
			2	2	外部引脚中断 ch.2
			1	1	外部引脚中断 ch.1
			0	0	外部引脚中断 ch.0
29	13	0x074	15	-	外部引脚中断 ch.23
			14	-	外部引脚中断 ch.22
			13	-	外部引脚中断 ch.21
			12	-	外部引脚中断 ch.20
			11	-	外部引脚中断 ch.19
			10	-	外部引脚中断 ch.18
			9	-	外部引脚中断 ch.17
			8	-	外部引脚中断 ch.16
			7	-	外部引脚中断 ch.15
			6	-	外部引脚中断 ch.14
			5	-	外部引脚中断 ch.13
			4	12	外部引脚中断 ch.12
			3	11	外部引脚中断 ch.11
			2	10	外部引脚中断 ch.10
			1	9	外部引脚中断 ch.9
			0	8	外部引脚中断 ch.8
30	14	0x078	1	-	双计时器中断 ch.2
			0	-	双计时器中断 ch.1
31	15	0x07C	2	-	MFS ch.0 状态中断
			1	47	MFS ch.0 传输中断
			0	46	MFS ch.0 接收中断
32	16	0x080	2	-	MFS ch.1 状态中断
			1	49	MFS ch.1 传输中断
			0	48	MFS ch.1 接收中断
33	17	0x084	2	-	MFS ch.2 状态中断
			1	51	MFS ch.2 传输中断
			0	50	MFS ch.2 接收中断
34	18	0x088	2	-	MFS ch.3 状态中断
			1	53	MFS ch.3 传输中断
			0	52	MFS ch.3 接收中断
35	19	0x08C	0	54	MFS ch.4 接收中断
36	20	0x090	1	-	MFS ch.4 状态中断
			0	55	MFS ch.4 传输中断
37	21	0x094	0	56	MFS ch.5 接收中断
38	22	0x098	1	-	MFS ch.5 状态中断
			0	57	MFS ch.5 传输中断

Exc. 号	IRQ 号	向量偏移	位	DSTC	异常和中断源
39	23	0x09C	12	-	DSTC ERINT 中断
			11	-	DSTC SWINT 中断
			10	-	智能卡（IC 卡）中断 ch.0
			9	-	HDMI-CEC 遥控接收中断 ch.0
			8~3	-	保留
			2	44	PPG 中断 ch.4
			1	43	PPG 中断 ch.2
			0	42	PPG 中断 ch.0
40	24	0x0A0	10	-	智能卡（IC 卡）中断 ch.1
			9	-	HDMI-CEC 遥控接收中断 ch.1
			8~6	-	保留
			5	-	实时时钟计数器中断
			4	45	计时计数器中断
			3	-	USB PLL 振荡稳定等待完成中断
			2	-	主 PLL 振荡稳定等待完成中断
			1	-	副时钟振荡稳定等待完成中断
			0	-	主时钟振荡稳定等待完成中断
41	25	0x0A4	4	-	ADC 单元 0 范围比较结果中断
			3	-	ADC 单元 0 结果比较中断
			2	-	ADC 单元 0 FIFO 溢出中断
			1	62	ADC 单元 0 扫描转换中断
			0	63	ADC 单元 0 优先转换中断
42	26	0x0A8	5	-	USB ch.0 主机 SOFIRQ 中断 USB ch.0 主机 CMPIRQ 中断
			4	-	USB ch.0 主机 DIRQ 中断 USB ch.0 主机 URIRQ 中断 USB ch.0 主机 RWKIRQ 中断 USB ch.0 主机 CNNIRQ 中断
			3	-	USB ch.0 从机 SPK 中断
			2	-	USB ch.0 从机 SUSP 中断 USB ch.0 从机 SOF 中断 USB ch.0 从机 BRST 中断 USB ch.0 从机 CONF 中断 USB ch.0 从机 WKUP 中断
			1	-	USB ch.0 从机端点 0 DRQO 中断
			0	-	USB ch.0 从机端点 0 DRQI 中断
43	27	0x0AC	5	-	LCD 控制器中断
			4~0	-	保留

Exc. 号	IRQ 号	向量偏移	位	DSTC	异常和中断源
44	28	0x0B0	15	41	MFT 单元 0 OCU 匹配检测中断 ch.5
			14	40	MFT 单元 0 OCU 匹配检测中断 ch.4
			13	39	MFT 单元 0 OCU 匹配检测中断 ch.3
			12	38	MFT 单元 0 OCU 匹配检测中断 ch.2
			11	37	MFT 单元 0 OCU 匹配检测中断 ch.1
			10		MFT 单元 0 OCU 匹配检测中断 ch.0
			9		MFT 单元 0 ICU 输入沿检测中断 ch.3
			8		MFT 单元 0 ICU 输入沿检测中断 ch.2
			7		MFT 单元 0 ICU 输入沿检测中断 ch.1
			6		MFT 单元 0 ICU 输入沿检测中断 ch.0
			5	31	MFT 单元 0 自由运行计时器零检测中断 ch.2
			4	30	MFT 单元 0 自由运行计时器零检测中断 ch.1
			3	29	MFT 单元 0 自由运行计时器零检测中断 ch.0
			2	28	MFT 单元 0 自由运行计时器峰值检测中断 ch.2
			1	27	MFT 单元 0 自由运行计时器峰值检测中断 ch.1
			0	26	MFT 单元 0 自由运行计时器峰值检测中断 ch.0
45	29	0x0B4	8	36	USB ch.0 从机端点 5 DRQ 中断
			7	35	USB ch.0 从机端点 4 DRQ 中断
			6	34	USB ch.0 从机端点 3 DRQ 中断
			5	33	USB ch.0 从机端点 2 DRQ 中断
			4	32	USB ch.0 从机端点 1 DRQ 中断
			3~0	-	保留
46	30	0x0B8	31~0	-	保留
47	31	0x0BC	27	-	闪存 RDY、HANG 中断
			26~16	-	保留
			15	-	基本计时器 ch.7 源 1 (IRQ1)
			14	-	基本计时器 ch.7 源 0 (IRQ0)
			13	-	基本计时器 ch.6 源 1 (IRQ1)
			12	22	基本计时器 ch.6 源 0 (IRQ0)
			11	-	基本计时器 ch.5 源 1 (IRQ1)
			10	-	基本计时器 ch.5 源 0 (IRQ0)
			9	-	基本计时器 ch.4 源 1 (IRQ1)
			8	21	基本计时器 ch.4 源 0 (IRQ0)
			7	20	基本计时器 ch.3 源 1 (IRQ1)
			6	19	基本计时器 ch.3 源 0 (IRQ0)
			5	18	基本计时器 ch.2 源 1 (IRQ1)
			4	17	基本计时器 ch.2 源 0 (IRQ0)
			3	16	基本计时器 ch.1 源 1 (IRQ1)
			2	15	基本计时器 ch.1 源 0 (IRQ0)
			1	14	基本计时器 ch.0 源 1 (IRQ1)
			0	13	基本计时器 ch.0 源 0 (IRQ0)

1.2 输入至 DSTC 的中断信号

Table 1-2 列出作为传输请求信号输入至 DSTC 的中断信号。表中编号对应 DSTC 的 DREQENB[n] 寄存器编号。

Table 1-2 输入至 DSTC 的中断信号列表

编号	中断信号名称
0	外部引脚中断 ch.0
1	外部引脚中断 ch.1
2	外部引脚中断 ch.2
3	外部引脚中断 ch.3
4	外部引脚中断 ch.4
5	外部引脚中断 ch.5
6	外部引脚中断 ch.6
7	外部引脚中断 ch.7
8	外部引脚中断 ch.8
9	外部引脚中断 ch.9
10	外部引脚中断 ch.10
11	外部引脚中断 ch.11
12	外部引脚中断 ch.12
13	基本定时器 ch.0 源 0 (IRQ0) 中断
14	基本定时器 ch.0 源 1 (IRQ1) 中断
15	基本定时器 ch.1 源 0 (IRQ0) 中断
16	基本定时器 ch.1 源 1 (IRQ1) 中断
17	基本定时器 ch.2 源 0 (IRQ0) 中断
18	基本定时器 ch.2 源 1 (IRQ1) 中断
19	基本定时器 ch.3 源 0 (IRQ0) 中断
20	基本定时器 ch.3 源 1 (IRQ1) 中断
21	基本定时器 ch.4 源 0 (IRQ0) 中断
22	基本定时器 ch.6 源 0 (IRQ0) 中断
23	MFT 单元 0 WFG 计时器 10 中断
24	MFT 单元 0 WFG 计时器 32 中断
25	MFT 单元 0 WFG 计时器 54 中断
26	MFT 单元 0 FRT ch.0 峰值检测中断
27	MFT 单元 0 FRT ch.1 峰值检测中断
28	MFT 单元 0 FRT ch.2 峰值检测中断
29	MFT 单元 0 FRT ch.0 零检测中断
30	MFT 单元 0 FRT ch.1 零检测中断
31	MFT 单元 0 FRT ch.2 零检测中断
32	USB ch.0 从机端点 1 DRQ 中断
33	USB ch.0 从机端点 2 DRQ 中断
34	USB ch.0 从机端点 3 DRQ 中断
35	USB ch.0 从机端点 4 DRQ 中断
36	USB ch.0 从机端点 5 DRQ 中断
37	MFT 单元 0 OCU ch.1 匹配检测中断
38	MFT 单元 0 OCU ch.2 匹配检测中断

编号	中断信号名称
39	MFT 单元 0 OCU ch.3 匹配检测中断
40	MFT 单元 0 OCU ch.4 匹配检测中断
41	MFT 单元 0 OCU ch.5 匹配检测中断
42	PPG ch.0 中断
43	PPG ch.2 中断
44	PPG ch.4 中断
45	计时计数器中断
46	MFS ch.0 接收中断
47	MFS ch.0 传输中断
48	MFS ch.1 接收中断
49	MFS ch.1 传输中断
50	MFS ch.2 接收中断
51	MFS ch.2 传输中断
52	MFS ch.3 接收中断
53	MFS ch.3 传输中断
54	MFS ch.4 接收中断
55	MFS ch.4 传输中断
56	MFS ch.5 接收中断
57	MFS ch.5 传输中断
58	-
59	-
60	-
61	-
62	A/D 转换器单元 0 扫描转换中断
63	A/D 转换器单元 0 优先转换中断

2. 寄存器

本节说明各寄存器详情。

寄存器表

缩写	寄存器名称	参考章节
EXC02MON	EXC02 批量读取寄存器	2.1
IRQ00MON	IRQ000 批量读取寄存器	2.2
IRQ01MON	IRQ001 批量读取寄存器	
IRQ02MON	IRQ002 批量读取寄存器	
IRQ03MON	IRQ003 批量读取寄存器	
IRQ04MON	IRQ004 批量读取寄存器	
IRQ05MON	IRQ005 批量读取寄存器	
IRQ06MON	IRQ006 批量读取寄存器	
IRQ07MON	IRQ007 批量读取寄存器	
IRQ08MON	IRQ008 批量读取寄存器	
IRQ09MON	IRQ009 批量读取寄存器	
IRQ10MON	IRQ010 批量读取寄存器	
IRQ11MON	IRQ011 批量读取寄存器	
IRQ12MON	IRQ012 批量读取寄存器	
IRQ13MON	IRQ013 批量读取寄存器	
IRQ14MON	IRQ014 批量读取寄存器	
IRQ15MON	IRQ015 批量读取寄存器	
IRQ16MON	IRQ016 批量读取寄存器	
IRQ17MON	IRQ017 批量读取寄存器	
IRQ18MON	IRQ018 批量读取寄存器	
IRQ19MON	IRQ019 批量读取寄存器	
IRQ20MON	IRQ020 批量读取寄存器	
IRQ21MON	IRQ021 批量读取寄存器	
IRQ22MON	IRQ022 批量读取寄存器	
IRQ23MON	IRQ023 批量读取寄存器	
IRQ24MON	IRQ024 批量读取寄存器	
IRQ25MON	IRQ025 批量读取寄存器	
IRQ26MON	IRQ026 批量读取寄存器	
IRQ27MON	IRQ027 批量读取寄存器	
IRQ28MON	IRQ028 批量读取寄存器	
IRQ29MON	IRQ029 批量读取寄存器	
IRQ30MON	IRQ030 批量读取寄存器	
IRQ31MON	IRQ031 批量读取寄存器	
IRQCMODE	中断因素向量重定位设置寄存器	2.3
RCINTSEL0	中断因素选择寄存器 0	2.4
RCINTSEL1	中断因素选择寄存器 1	2.5

2.1 EXC02 批量读取寄存器 (EXC02MON)

EXC02 批量读取寄存器 (EXC02MON) 可一次读取分配给异常 2 的所有中断（NMI 和硬件看门狗中断）。

寄存器配置

位	31							8
字段	保留							
属性	R							
初始值	0x000000							

位	7	6	5	4	3	2	1	0
字段	保留						HWINT	NMI
属性	R						R	R
初始值	000000						0	0

寄存器功能

[bit31:2] 保留：保留位

保留位读为 “0”。

[bit1] HWINT

值	描述
0	没有硬件看门狗计时器的中断请求。
1	已发出硬件看门狗计时器的中断请求。

[bit0] NMI

值	描述
0	没有 NMIX 外部引脚的中断请求。
1	已发出 NMIX 外部引脚的中断请求。

2.2 IRQxx 批量读取寄存器 (IRQxxMON)

IRQxx 批量读取寄存器表示 32 个寄存器（即： IRQ00MON – IRQ31MON ）。它们分别与 NVIC 的 IRQ00 – IRQ31 中断输入对应。逻辑“或”电路集合的中断信号状态可通过这些寄存器可读出。

参见配置表，检查此寄存器的哪一位分配给外设功能的哪一个中断。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	INT31	INT30	INT29	INT28	INT27	INT26	INT25	INT24
属性	R	R	R	R	R	R	R	R
初始值	0	0	0	0	0	0	0	0

位	23	22	21	20	19	18	17	16
字段	INT23	INT22	INT21	INT20	INT19	INT18	INT17	INT16
属性	R	R	R	R	R	R	R	R
初始值	0	0	0	0	0	0	0	0

位	15	14	13	12	11	10	9	8
字段	INT15	INT14	INT13	INT12	INT11	INT10	INT09	INT08
属性	R	R	R	R	R	R	R	R
初始值	0	0	0	0	0	0	0	

位	7	6	5	4	3	2	1	0
字段	INT07	INT06	INT05	INT04	INT03	INT02	INT01	INT00
属性	R	R	R	R	R	R	R	R
初始值	0	0	0	0	0	0	0	0

寄存器功能

[bit31:0] INT31 – INT00

操作	描述
写入	忽略写访问。
读取 0	分配给此位的中断未发生。
读取 1	发生分配给此位的中断。

2.3 中断因素向量重定位设置寄存器 (IRQCMODE)

中断因素向量重定位设置寄存器 (IRQCMODE) 选择根据“中断 (A)”章 Table 1-1 或根据本章 Table 1-1 分配中断因素向量。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	R/W															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留															IRQCMODE
属性	R/W															R/W
初始值	000000000000000															0

寄存器功能

[bit31:1] 保留：保留位
保留位读为 “0”。

[bit0] IRQCMODE:

位	描述
0	根据 “中断 (A)”一章的 Table 1-1 分配中断因素向量。
1	根据本章 Table 1-1 分配中断因素向量。

2.4 中断因素选择寄存器 0 (RCINTSEL0)

中断因素选择寄存器 0 (RCINTSEL0) 选择异常 19 至异常 22 的中断因素（IRQ03~IRQ06）。该寄存器在 IRQQCMODE:IRQCMODE 位为 "1" 时有效。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	INTSEL3								INTSEL2							
属性	R/W								R/W							
初始值	0x00								0x00							

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	INTSEL1								INTSEL0							
属性	R/W								R/W							
初始值	0x00								0x00							

寄存器功能

[bit31:24] INTSEL3:
上述位选择*异常 22 的中断因素 (IRQ06)。

[bit23:16] INTSEL2:
上述位选择*异常 21 的中断因素 (IRQ05)。

[bit15:8] INTSEL1:
上述位选择*异常 20 的中断因素 (IRQ04)。

[bit7:0] INTSEL0:
上述位选择*异常 19 的中断因素 (IRQ03)。

*: 有关中断因素的选择，详见 Table 2-1。

注意事项:

- *RCINTSEL0 选择的中断因素受 IRQ11~IRQ31 屏蔽。(IRQ11MON ~ IRQ31MON 寄存器中的可用位也被屏蔽。)*
- *选择中断因素时，确认从 INTSEL0 ~ INTSEL7 位选择的所有中断因素各不相同。*

2.5 中断因素选择寄存器 1 (RCINTSEL1)

中断因素选择寄存器 1 (RCINTSEL1) 选择异常 23 至异常 26 的中断因素（IRQ07~IRQ10）。该寄存器在 IRQQCMODE:IRQCMODE 位为 "1" 时有效。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	INTSEL7								INTSEL6							
属性	R/W								R/W							
初始值	0x00								0x00							

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	INTSEL5								INTSEL4							
属性	R/W								R/W							
初始值	0x00								0x00							

寄存器功能

[bit31:24] INTSEL7:
上述位选择*异常 26 的中断因素 (IRQ10)。

[bit23:16] INTSEL6:
上述位选择*异常 25 的中断因素 (IRQ09)。

[bit15:8] INTSEL5:
上述位选择*异常 24 的中断因素 (IRQ08)。

[bit7:0] INTSEL4:
上述位选择*异常 23 的中断因素 (IRQ07)。

*: 有关中断因素的选择，详见 Table 2-1。

注意事项:

- *RCINTSEL0 选择的中断因素受 IRQ11~IRQ31 屏蔽。(IRQ11MON ~ IRQ31MON 寄存器中的可用位也被屏蔽。)*
- *选择中断因素时，确认从 INTSEL0 ~ INTSEL7 位选择的所有中断因素各不相同。*

Table 2-1 中断因素选择

设置 RCINTSELx:INTSELx	中断因素
0x00	未选择中断因素。
0x01	外部中断 ch.0
0x02	外部中断 ch.1
0x03	外部中断 ch.2
0x04	外部中断 ch.3
0x05	外部中断 ch.4
0x06	外部中断 ch.5
0x07	外部中断 ch.6
0x08	外部中断 ch.7
0x09	外部中断 ch.8
0x0A	外部中断 ch.9
0x0B	外部中断 ch.10
0x0C	外部中断 ch.11
0x0D	基本计时器 ch.0 的 IRQ0/IRQ1
0x0E	基本计时器 ch.1 的 IRQ0/IRQ1
0x0F	基本计时器 ch.2 的 IRQ0/IRQ1
0x10	基本计时器 ch.3 的 IRQ0/IRQ1
0x11	基本计时器 ch.4 的 IRQ0/IRQ1
0x12	基本计时器 ch.5 的 IRQ0/IRQ1
0x13	基本计时器 ch.6 的 IRQ0/IRQ1
0x14	基本计时器 ch.7 的 IRQ0/IRQ1
0x15	MFS ch.0 的接收中断
0x16	MFS ch.1 的接收中断
0x17	MFS ch.2 的接收中断
0x18	MFS ch.3 的接收中断
0x19	MFT 单元 0 自由运行计时器 ch.0 的零检测中断
0x1A to 0xFF	保留

3. 使用注意事项

使用中断控制器时应注意以下各点：

- 按级别将外设功能的中断请求信号通知中断控制器。退出中断处理时，总是清除该中断的中断请求。
- 进程中有其它优先级更高的中断请求时，通知 NVIC 的外设功能中断请求会在 NVIC 内挂起。取消 NVIC 中挂起的中断请求时，清除外设功能的中断请求，并通过 NVIC 内安装的中断清除挂起寄存器（地址：0xE000E280 ~ 0xE000E29C）清除 NVIC 中挂起的中断请求。
- NMIX 引脚复用为通用端口。重置释放后，引脚的初始功能为通用端口，且 NMI 中断被屏蔽。使用 NMI 功能时，需通过端口设置使能 NMI 功能。详见“ 外部中断和 NMI 控制器”一章。
- 若使用由 DSTC 进行的 DMA 传输，将生成 DSTC 的传输结束中断 (HWINT[n])，而不是外设功能的中断。由于以上配置的原因，NVIC 产生外设功能的中断，且 DSTC 的传输结束中断跳转至同一中断向量。使用 DREQENB[n] 寄存器选择待处理的中断。
  但对于有些分别处理中断和向 DSTC 发出传输请求的外设功能（MFSI2S、HS-SPICNT、CAN-FD 及可编程 CRC），DSTC 的 DREQENB 寄存器设置确定是否执行 DMA 传输。在此情况下，外设功能的中断和 DSTC 的传输完成中断分别输入至 NVIC。
- 有关外设资源内具体事件检测寄存器和中断使能寄存器的关系，参见外设功能一章。

第 7-8 章：中断 (TYPE3)

本章说明中断控制器（**TYPE 3**）和外设中断请求（**TYPE 3**）。

1. 概述
2. 配置与功能
3. 中断列表
4. 寄存器
5. 使用注意事项

代码：9BFIRQC_B_FM0-T3-C03.0

1. 概述

本产品的 CPU 内核配有嵌套向量中断控制器 (NVIC)。 NVIC 支持预留系统异常和 32 个外设中断，还可设置 4 个中断优先级的优先顺序（通过一个内置 2 位寄存器进行）。本节说明微控制器内安装的外设功能发出的中断信号和 NVIC/DSTC 之间的关系。

2. 配置与功能

2.1 连接图

Figure 2-1 中断信号与 NVIC/DSTC 间的连接

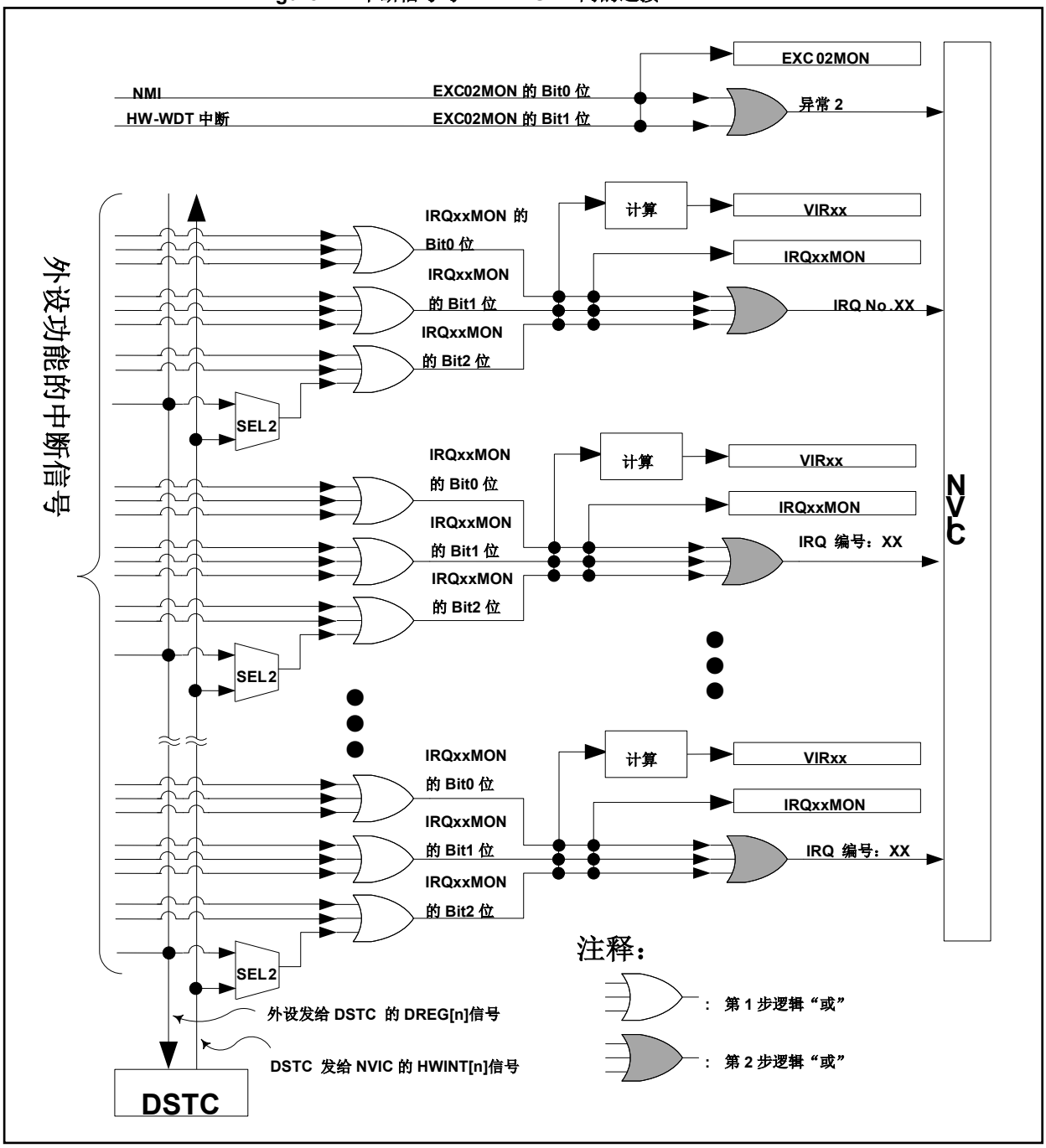


Figure 2-1 为 NVIC/DSTC 与外设功能中断信号输入间的连接图。详细说明如下：

2.2 NVIC (嵌套向量中断控制器)

NVIC（见 Figure 2-1）支持预留系统异常和 32 个外设中断。有关 NVIC，详见《Cortex-M0+ 技术参考手册》。《Cortex-M0+ 技术参考手册》将预留系统异常之外的异常定义为外部中断 (IRQ)。由于这里指的外部中断 (IRQ) 和微处理器外部输入引脚的外部中断存在差异，所以本文档将外部中断 (IRQ) 称为外设中断。

NVIC 的中断优先寄存器为 2 位配置，可设置 4 个中断优先级。

可使用 NVIC 中安装的系统处理程序优先级寄存器（地址：0xE000ED1C、0xE000ED20）分别设置预留系统异常 11、14 和 15 的优先级。

使用 NVIC 内安装的 IRQ 优先级寄存器（地址：0xE000E400 ~ 0xE000E41C）分别设置外设中断 16 ~ 47 的优先级。

NVIC 支持不可屏蔽中断 (NMI) 输入。

2.3 中断集合及批量读取寄存器

外设功能的中断信号（见 Figure 2-1）由第 1 阶输入逻辑“或”电路（见 Figure 2-1）和第 2 阶输入逻辑“或”电路（见 Figure 2-1）集合。然后集合的中断信号连接至 NVIC 的 32 个外设中断之一。参见 Table 3-1、Table 3-2 和 Table 3-3，检查哪一个外设功能中断信号分配给哪一个 NVIC 的 IRQ 输入。

由于中断信号通过逻辑“或”电路集合，因此多个源生成一个 NVIC 中断。生成中断后，通过读取中断批量读取寄存器（见 Figure 2-1 中 IRQxxMON）, CPU 可识别第 2 阶逻辑“或”集合的中断生成源。中断批量读取寄存器包含了 NVIC 的所有中断输入。支持 32 个寄存器 (IRQ00MON ~ IRQ31MON)。

CPU 不能通过读取中断批量读取寄存器识别通第 1 阶逻辑“或”集合的中断生成源。识别中断源需读取每个外设功能中的中断寄存器。

外部中断和 NMI 控制器的不可屏蔽中断信号 (NMI) 以及硬件看门狗计时器的 HW 看门狗中断通过 2 阶逻辑“或”集合，然后连接输入至 NVIC 的异常 2 。生成异常 2 后，可读取 EXC02 批量读取寄存器（Figure 2-1 中 EXC02MON）识别中断源，要么是 NMI ，要么是 HW 看门狗中断。

NMIX 引脚复用为通用端口。重置释放后，引脚的初始功能为通用端口，且 NMI 中断被屏蔽。使用 NMI 功能时，需通过端口设置使能 NMI 功能。详细信息，参见“I/O 端口”及“外部中断和 NMI 控制器”章节。

2.4 向量指示寄存器

发生中断时，CPU 可使用向量指示寄存器（见 Figure 2-1 中 VIRxx）快速转移中断操作。VIRxx 包括 32 个寄存器 (VIR00 – VIR32)分别对应 NVIC 的 IRQ00–IRQ31 输入。发生中断时，-CPU 能从 VIRxx 中读取例行程序的地址值。该地址值具体取决于第 2 阶逻辑“或”电路集合的中断信号状态。将中断处理程序的前列地址放入该地址域，CPU 可快速转移中断操作。发生中断 (IRQ00) 事件时 VIR00 的用法说明如下：

如 Table 2-1 所示，将 IRQ 中断处理程序的前列地址值放入地址域：0x0000 0040 -0x0000 00BC。同时将第 2 阶逻辑“或”电路集合的 Bit0、Bit1 和 Bit2 各中断对应的中断处理程序前列地址值放入地址区域：0x0000 00C0 -0x0000 01FC。

Table 2-1 中断向量表

地址	数据
0x0000 0000	栈指针初始值
0x0000 0004	异常 1：复位向量
0x0000 0008	异常 2：NMI/HW-WDT 处理程序前列地址
0x0000 000C	异常 3：硬故障处理程序前列地址
0x0000 0010 - 0x0000 0028	保留
0x0000 002C	异常 11：SVCall 处理程序前列地址
0x0000 0030 - 0x0000 0034	保留
0x0000 0038	异常 14：PendSV 处理程序前列地址
0x0000 003C	异常 15：SysTick 处理程序前列地址
0x0000 0040	IRQ00 处理程序前列地址
0x0000 0044	IRQ01 处理程序前列地址
....	....
0x0000 00B8	IRQ30 处理程序前列地址
0x0000 00BC	IRQ31 处理程序前列地址
0x0000 00C0	IRQ00 – bit0 处理程序前列地址
0x0000 00C4	IRQ01 – bit0 处理程序前列地址
....	....
0x0000 0138	IRQ30 – bit0 处理程序前列地址
0x0000 013C	IRQ31 – bit0 处理程序前列地址
0x0000 0140	IRQ00 – bit1 处理程序前列地址
0x0000 0144	IRQ01 – bit1 处理程序前列地址
....	....
0x0000 01B8	IRQ30 – bit1 处理程序前列地址
0x0000 01BC	IRQ31 – bit1 处理程序前列地址
0x0000 01C0	IRQ00 – bit2 处理程序前列地址
0x0000 01C4	IRQ01 – bit2 处理程序前列地址
....	....
0x0000 01F8	IRQ14 – bit2 处理程序前列地址
0x0000 01FC	IRQ15 – bit2 处理程序前列地址

Table 2-2 IRQ00 状态及 VIR00 读取值

IRQ00 输入信号状态			VIR00 读取值
Bit0 中断状态	Bit1 中断状态	Bit2 中断状态	
1	忽略	忽略	0x0000 00C0 + VIR_OFFSET
0	1	忽略	0x0000 0140 + VIR_OFFSET
0	0	1	0x0000 01C0 + VIR_OFFSET
0	0	0	未定义值

发生 IRQ00 时，IRQ00 中断处理程序启动。 CPU 读出此 IRQ00 中断处理程序中 VIR00 的地址值。从 VIR00 读出的地址值如 Table 2-2 所示。发生 Bit0 中断时，读出 0x0000 00C0。发生 Bit1 中断时，读出 0x0000 0140。同样，发生 Bit2 中断时，读出 0x0000 01C0。IRQ00 中断例程中，用户必须编程以便从 VIR00 寄存器读取地址并转移至其地址。这样就可快速执行中断处理程序的转移操作。

VIRxx 的读取值如 Table 3-1,Table 3-2 和 Table 3-3 中的 “VIR 值” 所述。除 VIR00 以外的 VIRxx 特性同 Table 2-2 所示。VIRxx 的使用注意事项说明如下：

CPU 必须按字（32 位宽）访问 VIRxx。若使用 VIRxx，中断判别操作的优先序列为固定序列，即 Bit0 > Bit1 > Bit2。

若未发生任何中断 (Bit0=Bit1=Bit2=0)，则 VIRxx 的读取值为未定义值。因此，发生中断后，CPU 应在清除中断信号前读取中断处理程序中的 VIRxx。读出 VIRxx 值并表示是否发生了中断。清除中断信号后，CPU 不能再次使用该中断处理程序的 VIRxx。

使用 VIR_OFFSET 寄存器将公共偏移值设置至 VIRxx。VIRxx 存在于 CM0+ 的单循环 I/O 区内，因此 CPU 可在 1 个周期内访问 VIRxx。

CPU 不能通过读取 VIRxx 识别第 1 阶逻辑 “或” 集合的中断因素。CPU 需读取各个外设功能内的中断寄存器，以识别中断因素。

2.5 DSTC 传输请求与传输完成的连接

外设功能发出的某些中断信号可用为发至 DSTC 的 DMA 传输请求信号。通过 DSTC 的 DREQENB[n] 寄存器设置，外设功能的中断信号被识别为 DMA 传输请求（Figure 2-1 所示外设向 DSTC 发出的 DREQ[n] 信号）。

输入至 NVIC 的中断信号由选择器电路（Figure 2-1 中 SEL2）选择。要么选择外设功能的中断信号或从 DSTC 输出的 DMA 传输完成通知信号（Figure 2-1 中 DSTC ~ NVIC 的 HWINT[n]）。根据 DREQENB[n] 寄存器设置切换 SEL2。

根据如图所示电路配置，若选择通过 DSTC 进行的 DMA 传输，将生成 DSTC 的传输完成中断，而不是外设的中断。DSTC 的传输完成中断可从 IRQxxMON 寄存器的对应位读取。此外，可应用 VIR 功能。有关每个外设功能的 DSTC 传输请求，详见相关外设功能章节。

2.6 DMA 传输确认信号的连接

有些外设功能模块中，向 DMA 传输至这些外设结束后必须清除传输请求信号（中断信号）。这类外设功能的传输请求信号通过 DSTC 清除。若在 SEL2 中选择通过 DSTC 进行的 DMA 传输，DSTC 的 DMA 传输确认信号（Figure 2-1 未说明）连接至某一外设功能。

2.7 DMA 传输停止信号的连接

DMA 传输停止请求信号从多功能串行单元（下文称为 "MFS"）输出。根据在 SEL2 中所做的选择，MFS（Figure 2-1 未说明）连接至 DSTC，具体说明如下：

如果在 SEL2 中如选择 DSTC 与 MFS 间的连接，DSTC 根据传输停止请求信号停止传输操作。DSTC 要在取反 MFS 的传输停止请求信号后才执行传输操作。MFS 的传输停止请求信号由逻辑 “或” 通过 DSTC 的传输结束中断（HWINT[n] 信号）集合并作为中断信号通知 NVIC。

3. 中断列表

本节列出输入至 NVIC 的异常源及中断源，以及通过 DSTC 的 DMA 传输所传输的中断。

3.1 异常及中断列表

Table 3-1,Table 3-2 及 Table 3-3 列出输入至 NVIC 的异常及中断。表中各列详细内容如下：

异常号

本列为 NVIC 的异常编号。

IRQ 号

本列为 NVIC 的外设中断编号（IRQ 编号 = 异常编号 - 16）

向量偏移

本列为发生中断时所引用的中断向量存储地址。引用 NVIC 中的描述值 + VTOR（向量表偏移寄存器）。

批量读取寄存器：名称

本列为批量读取寄存器的名称。本列中 "-" 表示没有该异常或中断的批量读取寄存器。

批量读取寄存器：位

本列为批量读取寄存器的哪一位分配给外设功能各异常和中断。本列中 "-" 表示没有适用于该异常或中断的批量读取寄存器。

VIR 值

本列为发生中断时从 VIRxx 读取的值。读取值为描述值 + VIR_OFFSET。本列中 "-" 表示没有 VIRxx。

异常或中断名称

本列为外设功能的异常或中断的名称。

如果批量读取寄存器一个位中描述多个中断源名称，表示这些中断由第 1 阶逻辑“或”电路集合。即使仅描述一个中断源名称，也可能存在有外设功能的逻辑“或”集合的多个中断源。详见各个外设功能。

从批量读取寄存器各位向 NVIC 发出的 IRQxx 中断信号由第 2 阶逻辑“或”电路生成。

产品中如果没有配备外设功能，则不会发生该外设功能的中断信号。因此对 IRQxxMON、VIRxx 和 NVIC 无效。

DSTC

本列表示支持 DSTC 的 DMA 传输。

本列中"数字" 按描述值表示支持 DSTC 的 DMA 传输。同时这个数字也表示 DSTC 的 DREQENB[n] 寄存器的位号。 DSTC 的 DREQENB[n] 寄存器设置决定 Figure 2-1 中 SEL2 的连接。

如果 DREQENB[n]=0，外设功能的中断信号连接至 NVIC，DSTC 则忽略该中断信号。

如果是 DREQENB[n]=1，外设功能的中断信号连接至 DSTC 作为 DMA 请求，而传输完成中断 (HWINT[n]) 连接至 NVIC。

"-" 按描述值表示不支持 DSTC 的 DMA 传输。

Table 3-1 异常及中断源列表 (表 1/3)

异常号	IRQ 号	向量偏移	批量读取寄存器		VIR 值	异常或中断名称	DSTC
			名称	位			
0	-	0x000	-	-	-	(栈指针初始值)	-
1	-	0x004	-	-	-	复位	-
2	-	0x008	EXC02MON	0	-	非屏蔽中断 (NMI)	-
				1	-	硬件看门狗计时器中断	-
3	-	0x00C	-	-	-	保留	-
4	-	0x010	-	-	-	保留	-
5	-	0x014	-	-	-	保留	-
6	-	0x018	-	-	-	保留	-
7	-	0x01C	-	-	-	保留	-
8	-	0x020	-	-	-	保留	-
9	-	0x024	-	-	-	保留	-
10	-	0x028	-	-	-	保留	-
11	-	0x02C	-	-	-	SVCall(监视器调用)	-
12	-	0x030	-	-	-	保留	-
13	-	0x034	-	-	-	保留	-
14	-	0x038	-	-	-	PendSV	-
15	-	0x03C	-	-	-	SysTick	-
16	0	0x040	IRQ00MON	0	0x0C0	CSV 的异常频率检测中断	-
				1	0x140	软件看门狗计时器中断	-
				2	0x1C0	低压检测 (LVD)中断	-
17	1	0x044	IRQ01MON	0	0x0C4	MFS ch.0 接收中断	0
				1	0x144	MFS ch.0 传送中断	1
				2	0x1C4	MFS ch.0 状态中断	-
18	2	0x048	IRQ02MON	0	0x0C8	MFS ch.1 接收中断	2
				1	0x148	MFS ch.1 传送中断	3
				2	0x1C8	MFS ch.1 状态中断	-
19	3	0x04C	IRQ03MON	0	0x0CC	保留	-
				1	0x14C	保留	-
				2	0x1CC	保留	-
20	4	0x050	IRQ04MON	0	0x0D0	MFS ch.3 接收中断	6
				1	0x150	MFS ch.3 传送中断	7
				2	0x1D0	MFS ch.3 状态中断	-
21	5	0x054	IRQ05MON	0	0x0D4	MFS ch.4 接收中断	8
				1	0x154	MFS ch.4 传送中断	9
				2	0x1D4	MFS ch.4 状态中断	-
22	6	0x058	IRQ06MON	0	0x0D8	保留	-
				1	0x158	保留	-
				2	0x1D8	保留	-
23	7	0x05C	IRQ07MON	0	0x0DC	MFS ch.6 接收中断	12
						I2CSLAVE 接收中断	48
				1	0x15C	MFS ch.6 传送中断	13
						I2CSLAVE 传送中断	49
				2	0x1DC	MFS ch.6 状态中断	-
						I2CSLAVE 状态中断	-

Table 3-2 异常及中断源列表 (表 2/3)

异常号	IRQ 号	向量偏移	批量读取寄存器		VIR 值	异常或中断名称	DSTC
			名称	位			
24	8	0x060	IRQ08MON	0	0x0E0	MFS ch.7 异常中断	14
				1	0x160	MFS ch.7 传送中断	15
				2	0x1E0	MFS ch.7 状态中断	-
25	9	0x064	IRQ09MON	0	0x0E4	A/D 转换器单元 0 优先转换中断	50
				1	0x164	A/D 转换器单元 0 描述转换中断	51
				2	0x1E4	A/D 转换器单元 0 FIFO 溢出中断	-
						A/D 转换器单元 0 转换结果比较中断	-
						A/D 转换器单元 0 范围比较结果中断	-
26	10	0x068	IRQ10MON	0	0x0E8	USB ch.0 设备端点 1 DRQ 中断	52
				1	0x168	USB ch.0 设备端点 2 DRQ 中断	53
				2	0x1E8	USB ch.0 设备端点 3 DRQ 中断	54
27	11	0x06C	IRQ11MON	0	0x0EC	USB ch.0 设备端点 4 DRQ 中断	55
				1	0x16C	USB ch.0 设备端点 5 DRQ 中断	56
				2	0x1EC	USB ch.0 设备端点 0 DRQI 中断	-
28	12	0x070	IRQ12MON	0	0x0F0	USB ch.0 设备端点 0 DRQO 中断	-
				1	0x170	USB ch.0 设备 SUSP 中断	-
						USB ch.0 设备 SOF 中断	-
						USB ch.0 设备 BRST 中断	-
						USB ch.0 设备 CONF 中断	-
						USB ch.0 设备 WKUP 中断	-
				2	0x1F0	USB ch.0 设备 SPK 中断	-
29	13	0x074	IRQ13MON	0	0x0F4	USB ch.0 主机 DIRQ 中断	-
						USB ch.0 主机 URIRQ 中断	-
						USB ch.0 主机 RWKIRQ 中断	-
						USB ch.0 主机 CNNIRQ 中断	-
				1	0x174	USB ch.0 主机 SOFIRQ 中断	-
						USB ch.0 主机 CMPIRQ 中断	-
				2	0x1F4	保留	-
30	14	0x078	IRQ14MON	0	0x0F8	主 PLL 振荡稳定等待完成中断	-
						主时钟振荡稳定等待完成中断	-
						副时钟振荡稳定等待完成中断	-
				1	0x178	保留	-
				2	0x1F8	保留	-
31	15	0x07C	IRQ15MON	0	0x0FC	看门狗计数器中断	57
				1	0x17C	实时计数器 (RTC) 中断	-
						双计时器 ch.1 中断	-
						双计时器 ch.2 中断	-
				2	0x1FC	保留	-

Table 3-3 异常及中断源列表 (表 3/3)

异常号	IRQ 号	向量偏移	批量读取寄存器		VIR 值	异常或中断名称	DSTC
			名称	位			
32	16	0x080	IRQ16MON	0	0x100	外部引脚中断 ch.0	16
				1	0x180	外部引脚中断 ch.1	17
33	17	0x084	IRQ17MON	0	0x104	外部引脚中断 ch.2	18
				1	0x184	外部引脚中断 ch.3	19
34	18	0x088	IRQ18MON	0	0x108	外部引脚中断 ch.4	20
				1	0x188	外部引脚中断 ch.5	21
35	19	0x08C	IRQ19MON	0	0x10C	外部引脚中断 ch.6	22
				1	0x18C	外部引脚中断 ch.7	23
36	20	0x090	IRQ20MON	0	0x110	外部引脚中断 ch.8	24
				1	0x190	保留	-
37	21	0x094	IRQ21MON	0	0x114	保留	-
				1	0x194	保留	-
38	22	0x098	IRQ22MON	0	0x118	外部引脚中断 ch.12	28
				1	0x198	外部引脚中断 ch.13	29
39	23	0x09C	IRQ23MON	0	0x11C	保留	-
				1	0x19C	外部引脚中断 ch.15	31
40	24	0x0A0	IRQ24MON	0	0x120	基本计时器 ch.0 源 0 (IRQ0) 中断	32
						基本计时器 ch.0 源 1 (IRQ1) 中断	33
				1	0x1A0	基本计时器 ch.4 源 0 (IRQ0) 中断	34
						基本计时器 ch.4 源 1 (IRQ1) 中断	35
41	25	0x0A4	IRQ25MON	0	0x124	基本计时器 ch.1 源 0 (IRQ0) 中断	36
						基本计时器 ch.1 源 1 (IRQ1) 中断	37
				1	0x1A4	基本计时器 ch.5 源 0 (IRQ0) 中断	38
						基本计时器 ch.5 源 1 (IRQ1) 中断	39
42	26	0x0A8	IRQ26MON	0	0x128	基本计时器 ch.2 源 0 (IRQ0) 中断	40
						基本计时器 ch.2 源 1 (IRQ1) 中断	41
				1	0x1A8	基本计时器 ch.6 源 0 (IRQ0) 中断	42
						基本计时器 ch.6 源 1 (IRQ1) 中断	43
43	27	0x0AC	IRQ27MON	0	0x12C	基本计时器 ch.3 源 0 (IRQ0) 中断	44
						基本计时器 ch.3 源 1 (IRQ1) 中断	45
				1	0x1AC	基本计时器 ch.7 源 0 (IRQ0) 中断	46
						基本计时器 ch.7 源 1 (IRQ1) 中断	47
44	28	0x0B0	IRQ28MON	0	0x130	CEC 接收/遥控接收 ch.0 中断	-
						CEC 传送 ch.0 中断	-
				1	0x1B0	CEC 接收/遥控接收 ch.1 中断	-
						CEC 传送 ch.1 中断	-
45	29	0x0B4	IRQ29MON	0	0x134	智能卡 ch.1 中断	-
				1	0x1B4	闪存 RDY/HANG 中断	-
46	30	0x0B8	IRQ30MON	0	0x138	DSTC SW 传输完成中断	-
				1	0x1B8	DSTC 错误中断	-
47	31	0x0BC	IRQ31MON	0	0x13C	保留	-
				1	0x1BC	保留	-

3.2 输入至 DSTC 的传输请求信号列表

Table 3-4 列出作为传输请求信号输入至 DSTC 的中断信号。表中编号对应 DSTC 的 DREQENB[n] 寄存器编号。

Table 3-4 输入至 DSTC 的中断信号列表

编号	DSTC 传输请求源
0	MFS ch.0 接收中断
1	MFS ch.0 传输中断
2	MFS ch.1 接收中断
3	MFS ch.1 传输中断
4	保留
5	保留
6	MFS ch.3 接收中断
7	MFS ch.3 传输中断
8	MFS ch.4 接收中断
9	MFS ch.4 传输中断
10	保留
11	保留
12	MFS ch.6 接收中断
13	MFS ch.6 传输中断
14	MFS ch.7 接收中断
15	MFS ch.7 传输中断
16	外部引脚中断 ch.0
17	外部引脚中断 ch.1
18	外部引脚中断 ch.2
19	外部引脚中断 ch.3
20	外部引脚中断 ch.4
21	外部引脚中断 ch.5
22	外部引脚中断 ch.6
23	外部引脚中断 ch.7
24	外部引脚中断 ch.8
25	保留
26	保留
27	保留
28	外部引脚中断 ch.12
29	外部引脚中断 ch.13

编号	DSTC 传输请求源
32	基本计时器 ch.0 源 0 (IRQ0) 中断
33	基本计时器 ch.0 源 1 (IRQ1) 中断
34	基本计时器 ch.4 源 0 (IRQ0) 中断
35	基本计时器 ch.4 源 1 (IRQ1) 中断
36	基本计时器 ch.1 源 0 (IRQ0) 中断
37	基本计时器 ch.1 源 1 (IRQ1) 中断
38	基本计时器 ch.5 源 0 (IRQ0) 中断
39	基本计时器 ch.5 源 1 (IRQ1) 中断
40	基本计时器 ch.2 源 0 (IRQ0) 中断
41	基本计时器 ch.2 源 1 (IRQ1) 中断
42	基本计时器 ch.6 源 0 (IRQ0) 中断
43	基本计时器 ch.6 源 1 (IRQ1) 中断
44	基本计时器 ch.3 源 0 (IRQ0) 中断
45	基本计时器 ch.3 源 1 (IRQ1) 中断
46	基本计时器 ch.7 源 0 (IRQ0) 中断
47	基本计时器 ch.7 源 1 (IRQ1) 中断
48	I2CSLAVE 接收中断
49	I2CSLAVE 传输中断
50	A/D 转换器单元 0 优先转换中断
51	A/D 转换器单元 0 扫描转换中断
52	USB ch.0 从机端点 1 DRQ 中断
53	USB ch.0 从机端点 2 DRQ 中断
54	USB ch.0 单元端点 3 DRQ 中断
55	USB ch.0 单元端点 4 DRQ 中断
56	USB ch.0 单元端点 5 DRQ 中断
57	计时计数器中断
58	保留
59	保留
60	保留
61	保留

编号	DSTC 传输请求源
30	保留
31	外部引脚中断 ch.15

编号	DSTC 传输请求源
62	保留
63	保留

4. 寄存器

本节分别详细说明本模块中的寄存器。

4.1 寄存器列表

Table 4-1 列出本模块的寄存器列表。

Table 4-1 寄存器列表

缩写	寄存器名称	参考章节
EXC02MON	EXC02 批量读取寄存器	4.2
IRQxxMON (xx is 00 to 31)	IRQxx 批量读取寄存器 （xx 表示 00 ~ 31）	4.3
VIR_OFFSET	VIR OFFSET 寄存器	4.4
VIRxx (xx is 00 to 31)	向量指寄存器 xx （xx 表示 00 ~ 31）	4.5
ODDPKS	USB ch.0 奇数包长 DMA 使能寄存器	4.6

4.2 EXC02 批量读取寄存器 (EXC02MON)

EXC02 批量读取寄存器 (EXC02MON) 可一次读取分配给异常 2 的所有中断（NMI 和硬件看门狗中断）。

寄存器配置

位	31							8
字段	保留							
属性	R							
初始值	0x000000							

位	7	6	5	4	3	2	1	0
字段	保留						HWINT	NMI
属性	R						R	R
初始值	000000						0	0

寄存器功能

[bit31:2] 保留：保留位

保留位读为 0。

[bit1] HWINT

操作	描述
写入	忽略写访问。
读取 0	发生硬件看门狗计时器中断请求
读取 1	未发生硬件看门狗计时器中断请求

[bit0] NMI

操作	描述
写入	忽略写访问。
读取 0	发生 NMIX 外部引脚中断请求
读取 1	未发生 NMIX 外部引脚中断请求

4.3 IRQxx 批量读取寄存器 (IRQxxMON)

IRQxx 批量读取寄存器表示 32 个寄存器（即：IRQ00MON – IRQ31MON）。它们分别与 NVIC 的 IRQ00 – IRQ31 中断输入对应。这些寄存器可读出第 2 阶逻辑“或”电路集合的中断信号的状态。

参见 Table 3-1、Table 3-2 和 Table 3-3，检查此寄存器的哪一位分配给外设功能的哪一个中断。

寄存器配置

位	31 … 3	2	1	0
字段	保留	INT2	INT1	INT0
属性	R	R	R	R
初始值	00000000000000000000000000000	0	0	0

寄存器功能

[bit31:1] 保留：保留位

保留位读为 0。

[bit2] INT2

操作	描述
写入	忽略写访问。
读取 0	未发生分配至此寄存器 Bit2 的中断。
读取 1	发生分配至此寄存器 Bit2 的中断。

[bit1] INT1

操作	描述
写入	忽略写访问。
读取 0	未发生分配至此寄存器 Bit1 的中断。
读取 1	发生分配至此寄存器 Bit1 的中断。

[bit0] INT0

操作	描述
写入	忽略写访问。
读取 0	未发生分配至此寄存器 Bit0 的中断。
读取 1	发生分配至此寄存器 Bit0 的中断。

仅读取中断批量读取寄存器，无法明确第 1 阶逻辑“或”集合的中断源。识别中断源时，CPU 需读取每个外设功能的中断寄存器。

4.4 VIR 偏移寄存器 (VIR_OFFSET)

VIR_OFFSET 寄存器用于定义VRxx的公共偏移值。发生中断时，VIRxx的读取值是此寄存器值和Table 3-1、Table 3-2 和 Table 3-3 中 VIR 值的总和。

寄存器配置

位	31	0
字段	VIR_OFFSET	
属性	R/W	
初始值	0x000000	

寄存器功能

[bit31:0] VIR_OFFSET

操作	描述
写入	定义 VIRx 的偏移值。
读取	读出寄存器值。

通常，此寄存器应设置为 NVIC 中 VTOR（向量表偏移寄存器）值相同。

此寄存器的 Bit[7:0] 必须总设置为 0x00。如需要改变此寄存器值，应在系统初始化时更改。不得在中断操作时更改此寄存器。

4.5 向量指示寄存器 xx (VIRxx)

发生中断时，CPU 可使用向量指示寄存器快速转移中断操作。VIRxx 包括 32 个寄存器 (VIR00 – VIR32)，分别对应 NVIC 的 IRQ00–IRQ31 输入。发生中断时，CPU 能读出 VIRxx 的例程地址值。该地址值由第 2 阶逻辑“或”电路集合的中断信号的状态决定。将中断处理程序的前列地址放入该地址域，CPU 可快速转移中断操作。

寄存器配置

位	31															0
字段	VIRxx															
属性	R															
初始值	xxxxxxxxxxxxxxxx															

寄存器功能

[bit31:0] VIRxx

操作	描述
写入	忽略写访问。
读取	读出第 2 阶逻辑“或”电路集合的中断信号状态对应的地址值

有关该寄存器的用法，参见 2.4 向量指示寄存器。VIRxx 的读取值如 Table 3-1,Table 3-2 和 Table 3-3 中的“VIR 值”所述。 VIR00 的特性示例见 Table 2-2。

CPU 必须按字（32 位宽）访问 VIRxx。若使用 VIRxx，中断判别操作的优先序列为固定序列，即 Bit0 > Bit1 > Bit2。

若未发生任何中断 (Bit0=Bit1=Bit2=0)，则 VIRxx 的读取值为未定义值。因此，发生中断后，CPU 应在清除中断信号前读取中断处理程序中的 VIRxx。读出 VIRxx 值并不表示是否发生中断。清除中断信号后，CPU 不能再次使用该中断处理程序的 VIRxx。

使用 VIR_OFFSET 寄存器将公共偏移值设置至 VIRxx。VIRxx 存在于 CM0+ 的单循环 I/O 区内，因此 CPU 可在 1 个周期内访问 VIRxx。

CPU 不能通过读取 VIRxx 明确第 1 阶逻辑“或”集合的中断源。CPU 需读取各个外设功能内的中断寄存器，以识别中断因素。

4.6 USB ch.0 奇数包长 DMA 使能寄存器 (ODDPKS)

如果在使用 DSTC 的 USB ch.0 自动传输时按 IN 方向传输数据，则仅最后一个包的最后数据的有效位宽在数据写入 USB 端点前被强制转换为 1 字节（8 位）。

寄存器配置

位	31	5	4	3	2	1	0
字段	保留		ODDPKS				
属性	R		R/W				
初始值	All 0		00000				

寄存器功能

[bit31:5] 保留：保留位

保留位写入 0 。保留位读为 0。

[bit4] ODDPKS4

操作	描述
写入 0	没有 DSTC 的 DMA 传输数据量的转换。
写入 1	若 DSTC 内的传输目标地址为 USB.EP5DT，将最后传输量转换为 1 字节。
读取	读出寄存器值。

[bit3] ODDPKS3

操作	描述
写入 0	没有 DSTC 的 DMA 传输数据量的转换。
写入 1	若 DSTC 内的传输目标地址为 USB.EP4DT，将最后传输量转换为 1 字节。
读取	读出寄存器值。

[bit2] ODDPKS2

操作	描述
写入 0	没有 DSTC 的 DMA 传输数据量的转换。
写入 1	若 DSTC 内的传输目标地址为 USB.EP3DT，将最后传输量转换为 1 字节。
读取	读出寄存器值。

[bit1] ODDPKS1

操作	描述
写入 0	没有 DSTC 的 DMA 传输数据量的转换。
写入 1	若 DSTC 内的传输目标地址为 USB.EP2DT，将最后传输量转换为 1 字节。
读取	读出寄存器值。

[bit0] ODDPKS0

操作	描述
写入 0	没有 DSTC 的 DMA 传输数据量的转换。
写入 1	若 DSTC 内的传输目标地址为 USB.EP1DT，将最后传输量转换为 1 字节。
读取	读出寄存器值。

注意事项：

- *该寄存器仅针对 USB 数据量自动传输模式下以 IN 方向传输 USB ch.0 数据时才有效。*

- *如果传输包的字节数为奇数，ODDPKS4、ODDPKS3、ODDPKS2、ODDPKS1 和 ODDPKS0 位均不得写入 1。*

5. 使用注意事项

使用中断控制器时应注意以下各点：

- 外设功能的中断请求信号按分级传感方法通知中断控制器。从中断处理中退出时，总是清除外设功能的中断请求信号。
- 进程中有其它优先级更高的中断请求时，通知 NVIC 的外设功能中断请求会在 NVIC 内挂起。取消 NVIC 中挂起的中断请求时，应从外设功能清除中断请求，然后通过 NVIC 中安装的中断清除挂起寄存器（地址：0xE000E280）清除 NVIC 中挂起的中断请求。
- NMIX 引脚复用为通用端口。重置释放后，引脚的初始功能为通用端口，且 NMI 中断被屏蔽。使用 NMI 功能时，需通过端口设置使能 NMI 功能。详细信息，参见 “I/O 端口” 及 “外部中断和 NMI 控制器” 章节。
- 若使用由 DSTC 进行的 DMA 传输，将生成 DSTC 的传输结束中断 (HWINT[n])，而不是外设功能的中断。由于以上配置的原因，NVIC 产生外设功能的中断，且 DSTC 的传输结束中断跳转至同一中断向量。使用 DREQENB[n] 寄存器选择待处理的中断。
- 有关外设资源内具体事件检测寄存器和中断使能寄存器的关系，参见外设功能一章。

第 8 章：外部中断及 NMI 控制段

本章说明外部中断和 **NMI** 控制段的功能和操作。

1. 概述
2. 框图
3. 操作及设置步骤示例
4. 寄存器

代码：9BFEXTINT-FM0-C03.0_FW12-E1.04

1. 概述

外部中断和 NMI 控制段具有以下功能特性：

- 最大安装 32 个外部中断输入引脚和 1 个 NMI 输入引脚。
- 可从下列五种类型中选择外部中断检测条件：
  - 高电平
  - 低电平
  - 上升沿
  - 下降沿
  - 上升沿和下降沿（仅 TYPE 3-M0+产品）
- 可使用外部中断输入或 NMI 输入从待机模式唤醒。
- TYPE 3-M0+产品中可使能/禁用 NMI 中断。

2. 框图

以下为外部中断和 NMI 控制段框图。

Figure 2-1 外部中断和 NMI 控制段框图（用于 TYPE 1/2-M0+产品）

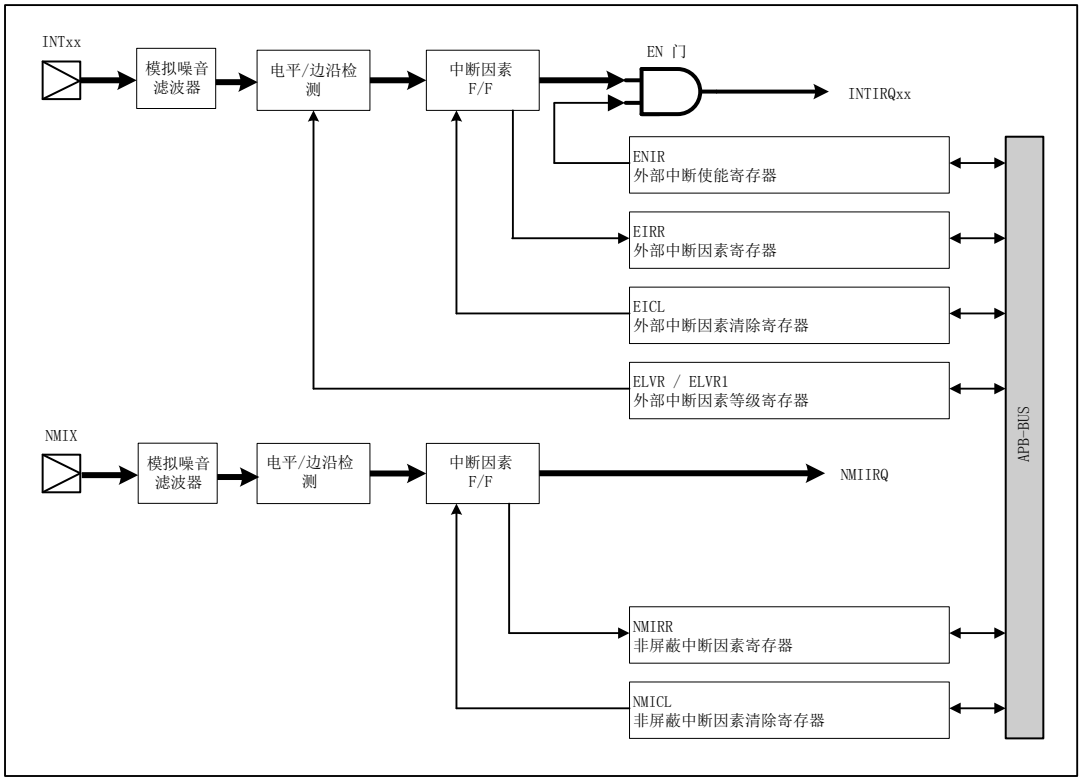
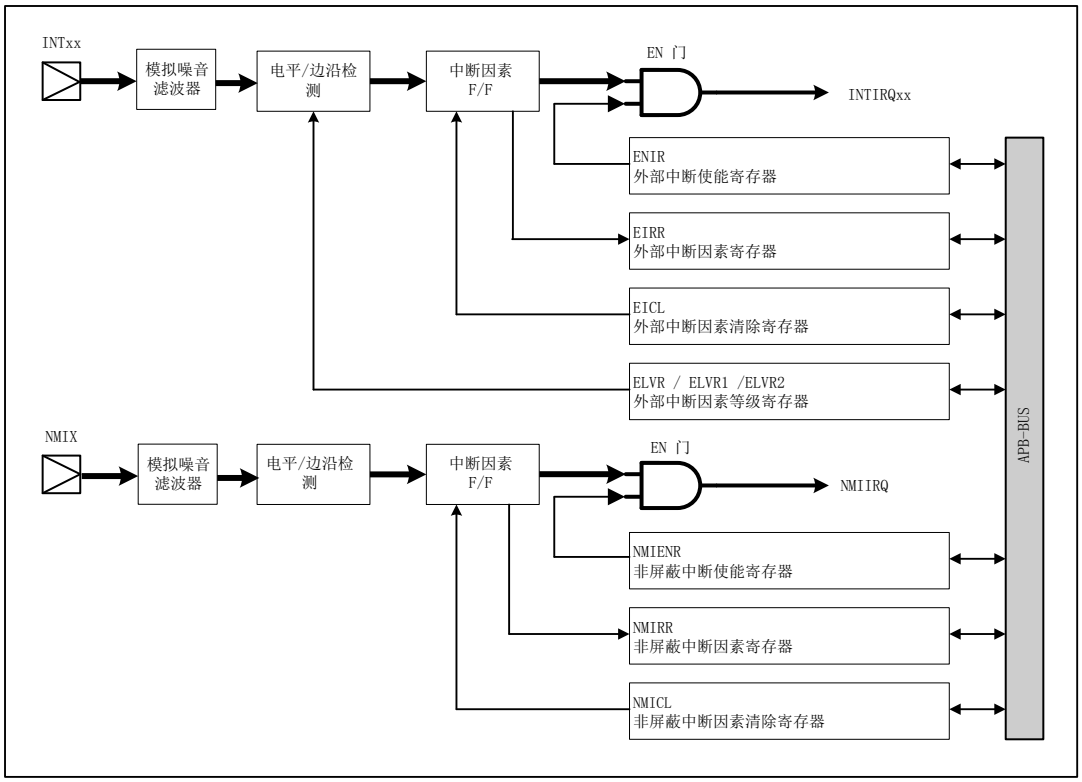


Figure 2-2 外部中断和 NMI 控制段框图（用于 TYPE 3-M0+产品）



3. 操作及设置步骤示例

本节说明操作和设置步骤示例。

3.1 外部中断控制段操作
3.2 NMI 控制段操作
3.3 从计时器、停止和 RTC 模式唤醒

3.1 外部中断控制段操作

本节说明外部中断控制段的操作。

外部中断控制段操作概述

外部中断控制段按照下面所述的序列将外部中断请求输出至中断控制器。

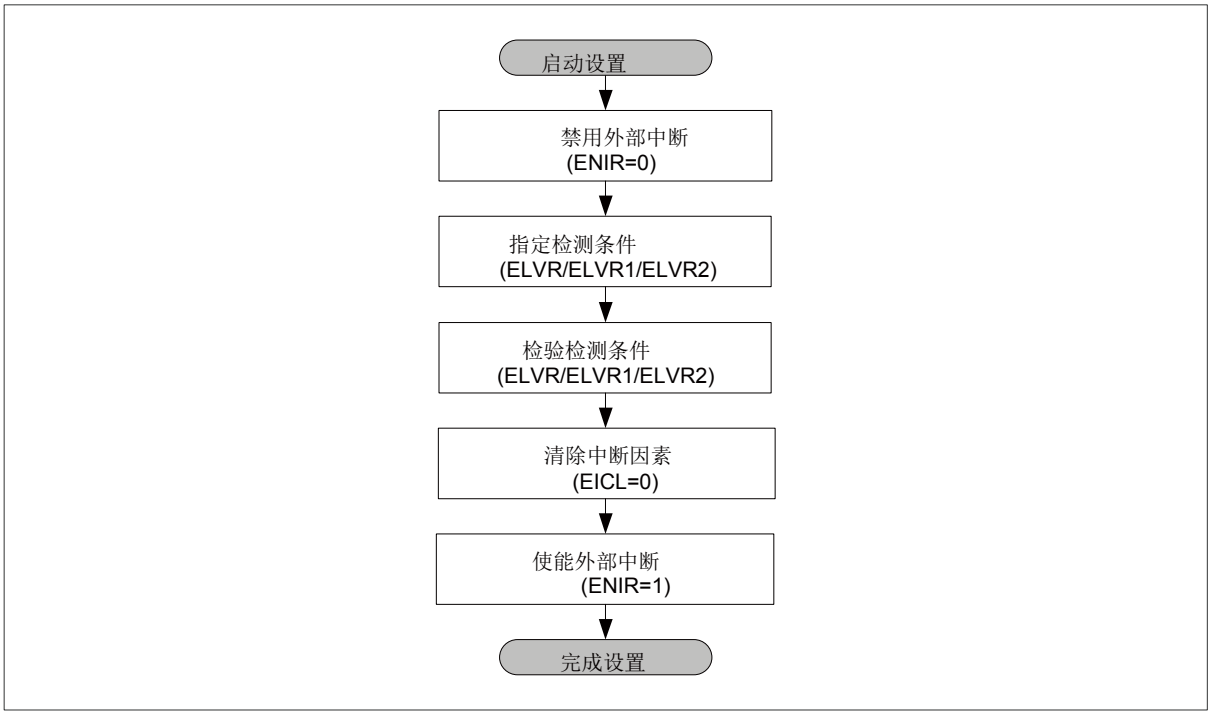
1. 将信号输入 INTxx 引脚，检测外部中断电平寄存器(ELVR/ELVR1/ELVR2)中指定的边沿或电平。待检测的边沿或电平可从下列五种类型中选择：
   高电平、低电平、上升沿、下降沿以及上升沿和下降沿同时检测
   其中，只有 TYPE 3-M0+产品支持“上升沿和下降沿同时检测”。）
2. 检测到的中断输入保持在中断因素 F/F 中。
   通过外部中断因素寄存器 (EIRR) 读取。
3. 如果通过外部中断使能寄存器 (ENIR) 禁用外部中断，则发送至中断寄存器的外部中断请求 (INTIRQxx) 生效。
4. 保持的中断因素通过外部中断因素清除寄存器 (EICL) 清除，向中断控制器发送外部中断请求 (INTIRQxx) 取反。

初始设置步骤

按下列步骤进行外部中断设置：

1. 通过外部中断使能寄存器 (ENIR=0) 禁用外部中断。
2. 通过外部中断因素电平寄存器 (ELVR/ELVR1/ELVR2) 指定检测条件（有效边沿或电平）
3. 读取外部中断因素电平寄存器（ELVR/ELVR1/ELVR2 之一）。
4. 通过外部中断因素清除寄存器清除外部中断因素 (EICL=0) 。
5. 通过外部中断使能寄存器禁用外部中断 (ENIR=1) 。

Figure 3-1 外部中断设置步骤

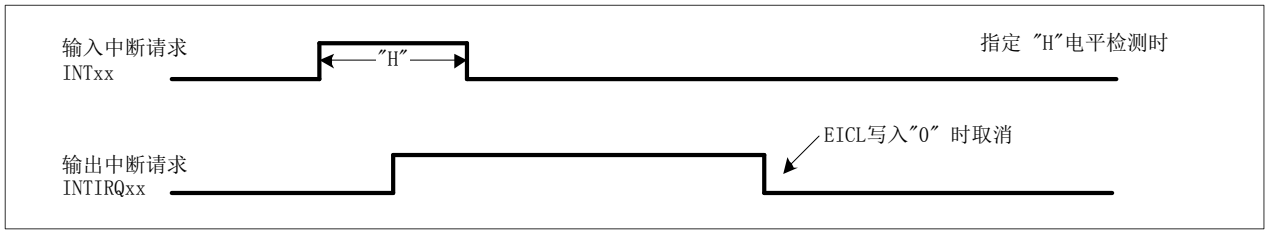


设置和清除外部中断请求

当检测到外部中断时，中断因素保持在外部中断因素寄存器 (EIRR) 中。Figure 3-2 所示为检测条件设置为高电平时的波形。当外部中断检测条件设置为 H 或 L 电平时，即使取反外部中断请求输入 (INTxx) ，中断因素仍然保持在外部中断因素寄存器 (EIRR) 中。因此，发送至中断控制器的外部中断请求 (INTIRQxx) 保持有效。当收到中断后，CPU 应通过外部中断因素清除寄存器 (EICL=0) 清除外部中断因素寄存器 (EIRR)。

当选择电平检测条件以及 INTxx 引脚输入生效时，即使 CPU 通过外部中断因素清除寄存器 (EICL) 清除外部中断因素寄存器 (EIRR)，本位也会重新设置为 1。

Figure 3-2 清除中断因素

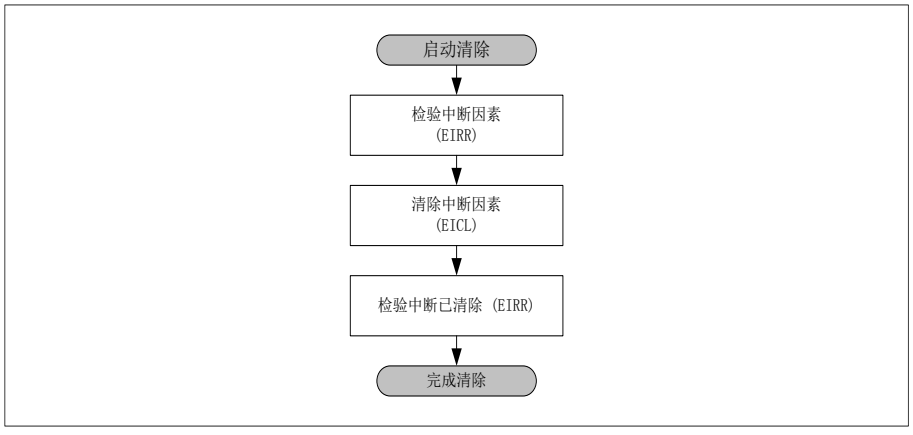


外部中断因素清除步骤

执行下列几步操作取消外部中断请求：

1. 读取外部中断因素寄存器 (EIRR) ，检查中断因素。
2. 在外部中断因素清除寄存器 (EICL) 的对应位写入 "0" 清除中断因素。
3. 读取外部中断因素寄存器 (EIRR) ，检查该中断因素是否清除。

Figure 3-3 清除外部中断请求



注意事项：

- *进程中有其它优先级更高的中断请求时，通知 NVIC 的外设功能中断请求会在 NVIC 内挂起。取消 NVIC 中挂起的中断请求时，应从外设功能清除中断请求，然后通过 NVIC 中安装的中断清除挂起寄存器（地址：0xE000E280）清除 NVIC 中挂起的中断请求。*

3.2 NMI 控制段操作

本节说明 NMI 控制段的操作。

NMI 控制段操作概述

当输入至 NMI 输入引脚 (NMIX) 的信号中检测到边沿或电平时，NMI 控制段会向 CPU 输出一个 NMI 中断请求 (NMIIRQ)。

检测下列边沿或电平（此条件不可更改）。

- 运行模式 : 下降沿
- 休眠模式 : 下降沿
- 计时器模式 : L 电平
- RTC 模式 : L 电平
- 停止模式 : L 电平
- 深度待机模式： NMI 请求不可用于此模式。

注意事项：

- *NMI 请求不可用于从深度待机模式唤醒。但是，NMIX 输入引脚和 WKUP 输入引脚复用同一输入引脚。因此，NMI 请求可用于通过 WKUP 输入引脚唤醒。详细信息参见第 6 章“低功耗模式”的“5. 深度待机模式下的操作”。*

NMIENR（NMI 使能寄存器）

TYPE 3-M0+产品中，NMI 请求可通过 NMIENR (NMI 使能寄存器) 使能或禁用。

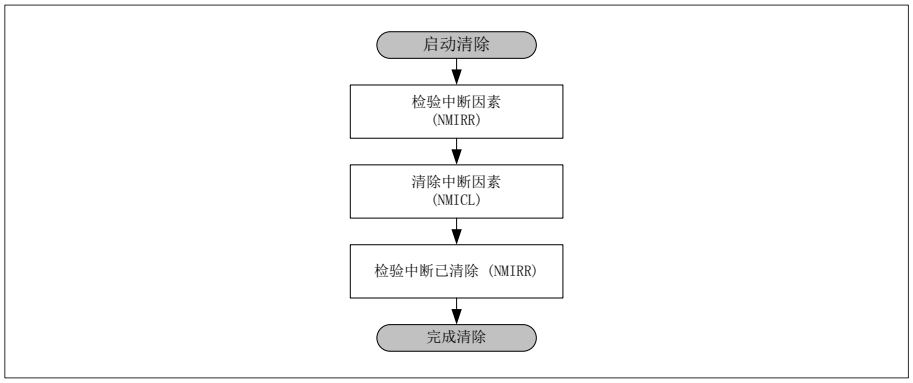
取消 NMI 请求

取消 NMI 请求时，按照清除外部中断请求相同的方法清除请求寄存器。

执行下列几步操作取消 NMI 中断请求：

1. 读取 NMI 因素寄存器 (NMIRR)，检查中断因素。
2. 在 NMI 因素清除寄存器 (NMICL) 的对应位写入 "0"，将其清除。
3. 读取 NMI 因素寄存器 (NMIRR)，检查中断因素是否已清除。

Figure 3-4 取消 NMI 请求



3.3 从计时器、停止和 RTC 模式唤醒

本节说明计时器、停止和 RTC 模式下的唤醒。

概述

外部中断和 NMI 请求可用于从计时器、停止和 RTC 模式唤醒。
在这些模式下，首先异步输入信号至引脚 INTxx 或 NMIX，设备可从这些模式唤醒至运行模式。

切换至计时器、停止和 RTC 模式前的设置

使用外部中断请求时，在切换至这些模式之前应指定用于从这些模式唤醒的引脚，并指定有效的检测电平。

- 用于从这些模式唤醒的引脚： 外部请求输出使能 (ENIR = 1)。
- 不用于从这些模式唤醒的引脚： 外部请求输出禁用 (ENIR = 0)。

使用 NMI 请求时，只检测低电平，不可选择检测条件。TYPE 3-M0+产品应使能 NMIENR 寄存器。

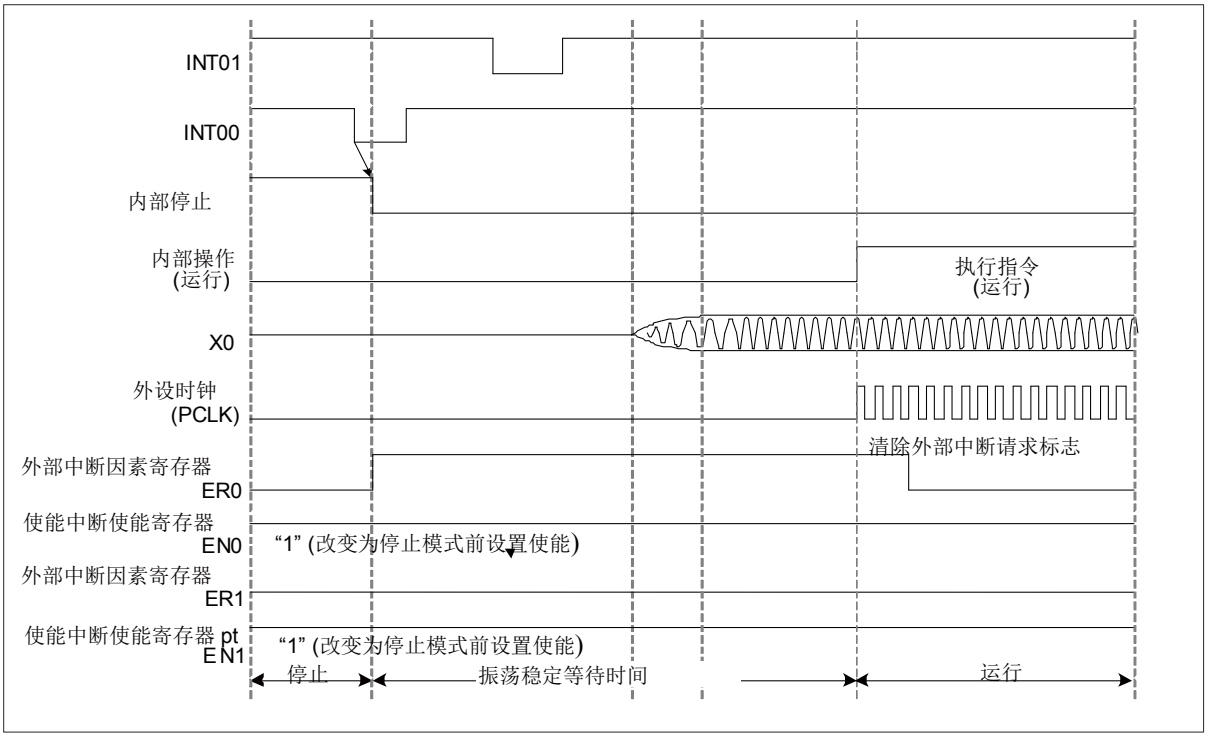
从计时器、停止及 RTC 模式唤醒

对于外部中断请求，如果用于从这些模式唤醒的引脚中检测到预先指定的有效电平，则设备会从这些模式唤醒。
使用 NMI 请求唤醒时，如果在停止模式检测到低电平，设备会从这些模式唤醒。

从计时器、停止及 RTC 模式唤醒时的注意事项

任何其他外部中断请求要在这些模式释放后振荡稳定等待期结束时才能被识别。
(对于 Figure 3-5 中的 INT01，任何外部中断请求都不能识别。)
因此，如果要在释放这些模式后输入外部中断，应在振荡稳定等待期结束后再输入外部中断请求。

Figure 3-5 从计时器、停止和 RTC 模式唤醒



4. 寄存器

本节列出寄存器列表。

寄存器列表

以下为外部中断和 NMI 控制段的寄存器列表。

Table 4-1 外部中断和 NMI 控制段的寄存器

缩写	寄存器名称	参考章节
ENIR	外部中断使能寄存器	4.1
EIRR	外部中断因素寄存器	4.2
EICL	外部中断因素清除寄存器	4.3
ELVR	外部中断因素电平寄存器	4.4
ELVR1	外部中断因素电平寄存器 1	4.5
NMIRR	非屏蔽中断因素寄存器	4.6
NMICL	非屏蔽中断因素清除寄存器	4.7
ELVR2	外部中断因素电平寄存器 2	4.8
NMIENR	非屏蔽中断使能寄存器	4.9

4.1 外部中断使能寄存器 (ENIR)

ENIR 寄存器用于控制屏蔽外部中断请求输出。

寄存器配置

位	31															16
字段	EN[31:16]															
属性	R/W															
初始值	0x0000															

位	15															0
字段	EN[15:0]															
属性	R/W															
初始值	0x0000															

寄存器功能

[bit31:0] EN31 ~ EN0：外部中断使能位

EN31 ~ EN0 位对应 INT31 ~ INT00 引脚。

产品规格未定义引脚对应的位应设置为 0。

位	描述
写入 0	禁用相关位对应的 INTx 引脚输出外部中断请求。
写入 1	使能相关位对应的 INTx 引脚输出外部中断请求。
读取	读出寄存器值。

此功能使能与此寄存器中设置为 1 的位所对应的中断请求输出，并向中断控制器输出中断请求。设置为 0 的位所对应的引脚保持中断因素，但不向中断控制器输出中断请求。

4.2 外部中断因素寄存器 (EIRR)

EIRR 寄存器指示是否检测到外部中断请求。

寄存器配置

位	31															16
字段	ER[31:16]															
属性	R															
初始值	0xXXXX															

位	15															0
字段	ER[15:0]															
属性	R															
初始值	0xXXXX															

寄存器功能

[bit31:0] ER31 ~ ER0：外部中断请求检测位

ER31 ~ ER0 位对应 INT31 ~ INT00 引脚。

产品规格未定义引脚的对应位是不确定的。

位	描述
读取 0	未检测到相关位对应 INTx 引脚的外部中断请求。
读取 1	检测到相关位对应 INTx 引脚的外部中断请求。
写入	操作无效

注意事项：

- *当选择电平检测条件以及 INTxx 引脚输入生效时，即使 CPU 通过外部中断因素清除寄存器 (EICL) 清除外部中断因素寄存器 (EIRR)，本位也会重新设置为 1。*
- *初始化时，外部中断因素寄存器 (EIRR)的位可设置为 1。在将 GPIO 设置至外部中断引脚并设置 ELVR/ELVR1/ELVR2 后，清除外部中断因素寄存器 (EIRR)。*

4.3 外部中断因素清除寄存器 (EICL)

EICL 寄存器用于清除保持的中断因素。

寄存器配置

位	31															16
字段	ECL[31:16]															
属性	W															
初始值	0xFFFF															

位	15															0
字段	ECL[15:0]															
属性	W															
初始值	0xFFFF															

寄存器功能

[bit31:0] ECL31 ~ ECL0：外部中断因素清除位

ECL31 ~ ECL0 位对应 INT31 ~ INT00 引脚。

不能在产品规格未定义引脚的对应位写入 0。

位	描述
写入 0	清除相关位对应 INTx 引脚的外部中断因素。
写入 1	操作无效
读取	总读为 1。

4.4 外部中断因素电平寄存器 (ELVR)

ELVR 用于选择作为外部中断请求而被检测到的信号的电平或边沿。

寄存器配置

位	31															16
字段	LB15	LA15	LB14	LA14	LB13	LA13	LB12	LA12	LB11	LA11	LB10	LA10	LB9	LA9	LB8	LA8
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位	15															0
字段	LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4	LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

寄存器功能

[bit31:0] LA15 ~ LA0 或 LB15 ~ LB0：外部中断请求检测电平选择位

LA15 ~ LA0 或 LB15 ~ LB0 位以 2 位 (LA 和 LB) 方式对应 INT15 ~ INT00 引脚。

产品规格未定义引脚的对应位应设置为 00。

当检测到此位选择的边沿或电平时，会被识别为外部中断请求。

	LBx	LAx	描述
写入	0	0	检测低电平。
	0	1	检测高电平。
	1	0	检测上升沿。
	1	1	检测下降沿。
读取	-	-	读出寄存器值。

注意事项：

- *当重新写入 ELVR/ELVR1/ELVR2 以修改检测条件时，可能会发生无效中断因素。为了避免发生无效中断因素，按照 Figure 3-1 所示步骤修改检测条件。*
- *检测 ELVR/ELVR1/ELVR2 中指定的边沿或电平时，至少需要 3T (T: PCLK 周期)的脉冲宽度。当输入的信号不能达到此脉冲宽度要求时，则不能保证操作正确。*

4.5 外部中断因素电平寄存器 1 (ELVR1)

ELVR1 用于选择作为外部中断请求而被检测到的信号的电平或边沿。

寄存器配置

位	31															16
字段	LB31	LA31	LB30	LA30	LB29	LA29	LB28	LA28	LB27	LA27	LB26	LA26	LB25	LA25	LB24	LA24
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位	15															0
字段	LB23	LA23	LB22	LA22	LB21	LA21	LB20	LA20	LB19	LA19	LB18	LA18	LB17	LA17	LB16	LA16
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

寄存器功能

[bit31:0] LA31 ~ LA16 或 LB31 ~ LB16：外部中断请求检测电平选择位

LA31 ~ LA16 或 LB31 ~ LB16 以 2 位 (LA 和 LB) 方式对应 INT31 ~ INT16 引脚。

产品规格未定义引脚的对应位应设置为 00。

当检测到此位选择的边沿或电平时，会被识别为外部中断请求。

	LBx	LAx	描述
写入	0	0	检测低电平。
	0	1	检测高电平。
	1	0	检测上升沿。
	1	1	检测下降沿。
读取	-	-	读出寄存器值。

注意事项：

- *当重新写入 ELVR/ELVR1/ELVR2 以修改检测条件时，可能会发生无效中断因素。为了避免发生无效中断因素，按照 Figure 3-1 所示步骤修改检测条件。*
- *检测 ELVR/ELVR1/ELVR2 中指定的边沿或电平时，至少需要 3T (T: PCLK 周期)的脉冲宽度。当输入的信号不能达到此脉冲宽度要求时，则不能保证操作正确。*

4.6 非屏蔽中断因素寄存器 (NMIRR)

NMIRR 寄存器用于检测是否已检测到非屏蔽中断 (NMI) 请求。

寄存器配置

位	15	1	0
字段	保留		NR
属性	-		R
初始值	-		0

寄存器功能

[bit15:1] 保留：保留位

读取值未定义。

写入模式时无效。

[bit0] NR： NMI 中断请求检测位

NR 位对应 NMIX 引脚。

位	描述
读取 0	未检测到 NMI 中断请求。
读取 1	检测到 NMI 中断请求。
写入	操作无效

注意事项：

若为 TYPE 1-M0+、TYPE 2-M0+ 产品

- *当映射至 NMI 输入引脚的 I/O 端口从 NMI 改变为映射至 GPIO 或其他外设功能（写入 EPFR00.NMIS = 1）时，I/O 端口的输入电平应保持高电平并改变 I/O 端口。若 I/O 端口被选为映射至 GPIO 或其他外设功能，内部 NMI 信号将保持在高电平。因此，当 I/O 端口输入电平为低电平时，I/O 端口从 GPIO 或其他外设功能改变为 NMI 会使内部 NMI 信号从高电平变为低电平。于是，会检测到下降沿，发生 NMI 请求。*

若为 TYPE 3-M0+ 产品

- *当映射至 NMI 输入引脚的 I/O 端口改变为从 NMI 至 GPIO 或其他外设功能（写入 EPFR00.NMIS = 1）时，应改变为通过非屏蔽中断使能寄存器 (NMIENR=0) 禁用 NMI 的状态。改变 GPIO 之后，通过 NMICL 清除 NMIRR。同时通过(NMIENR=1) 使能 NMI。*

4.7 非屏蔽中断因素清除寄存器(NMICL)

NMICL 寄存器用于清除保持的中断因素。

寄存器配置

位	15	1	0
字段	保留		NCL
属性	-		R/W
初始值	-		1

寄存器功能

[bit15:1] 保留：保留位

读取值未定义。

写入模式时无效。

[bit0] NCL： NMI 中断因素清除位

NCL 位对应 NMIX 引脚。

位	描述
写入 0	清除 NMI 中断因素。
写入 1	操作无效
读取	总读为 1。

4.8 外部中断因素电平寄存器 2 (ELVR2)

ELVR2 用于选择作为外部中断请求而被检测到的信号边沿和电平。只有 TYPE 3-M0+产品配有此寄存器。TYPE 1-M0+和 2-M0+产品不支持这个检测条件。

寄存器配置

位	31															16
字段	LC31	LC30	LC29	LC28	LC27	LC26	LC25	LC24	LC23	LC22	LC21	LC20	LC19	LC18	LC17	LC16
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位	15															0
字段	LC15	LC14	LC13	LC12	LC11	LC10	LC9	LC8	LC7	LC6	LC5	LC4	LC3	LC2	LC1	LC0
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

寄存器功能

[bit31:0] LC31 ~ LC0：外部中断请求检测电平选择位

LC31 ~ LC0 位对应 INT31 ~ INT0 引脚。

产品规格未定义引脚的对应位应设置为 0。

当检测到上升沿或下降沿时，都会被识别为外部中断请求。

	LCx	描述
写入	0	检测 ELVR 和 ELVR1 选择的边沿或电平。
	1	检测上升沿和下降沿。 ELVR/ELVR1 的设置被忽略。
读取	-	读出寄存器值。

注意事项：

- *当重新写入 ELVR/ELVR1/ELVR2 以修改检测条件时，可能会发生无效中断因素。为了避免发生无效中断因素，按照 Figure 3-1 所示步骤修改检测条件。*
- *检测 ELVR/ELVR1/ELVR2 中指定的边沿或电平时，至少需要 3T (T: PCLK 周期)的脉冲宽度。当输入的信号不能达到这个脉冲宽度要求时，则不能保证操作正确。*

4.9 非屏蔽中断使能寄存器 (NMIENR)

NMIENR 寄存器用于使能或禁用 NMI 中断。只有 TYPE 3-M0+产品配有此寄存器。TYPE 1-M0+和 2-M0+产品不支持这个功能。

寄存器配置

位	7	1	0
字段	保留		NE0
属性	-		R/W
Initial value	-		0

寄存器功能

[bit7:1] 保留：保留位

读取值未定义。

写入模式时无效。

[bit0] NE0： NMI 使能位

此位用于使能或禁用 NMI 中断。

位	功能
写入 0	禁用 NMI 中断 [初始值]
写入 1	使能 NMI 中断
读取	读取当前设定值

注意事项：

- *使能 NMI 中断前应通过 NMIRR 寄存器检查中断因素并用 NMICL 寄存器将其清除。如果未清除中断因素，则有可能在使能 NMI 中断时发生 NMI 中断。*
- *此寄存器的初始值为 “0”，NMI 禁用。*

第 9 章：DMAC

本章说明 **DMAC**。

1. **DMAC 概述**
2. **DMAC 配置**
3. **DMAC 功能及操作**
4. **DMAC 控制**
5. **DMAC 寄存器**
6. **使用注意事项**

代码：9BFDMAC_FM0-C03.0_MHDMAC-E01.0

1. DMAC 概述

DMAC（直接内存访问控制器）功能块可以不通过 CPU 高速传输数据。使用 DMAC 能提高系统性能。

DMAC 概述

- DMAC 配有独立于 CPU 总线的总线，所以即便是在使用 CPU 总线的时候，DMAC 也可进行传输操作。
- 最多由 8 条通道组成，能执行 8 种相互独立的 DMA 传输。
- 可设置传输目标地址、传输源地址、传输数据大小、传输请求源以及传输模式，并能控制各通道的传输操作启动、传输的强行终止以及传输的暂停。
- 可控制所有通道批量传输的启动、强行终止及暂停。
- 多通道同时操作时，可用固定方法或循环方法选择操作通道的优先级。
- 支持使用外设中断信号的硬件 DMA 传输。
- 遵从系统总线 (AHB)，支持 32 位地址空间 (4GB)。

各通道功能概述

- 可递增或固定传输源和传输目标地址。
- 具有传输源和传输目标地址重装功能（即传输完成后回到原设置值的功能）。
- 传输数据的大小可以从以下三种规格中选择：

  传输数据宽度：（从字节/半字/字中选择）
  设置块数：（从 1 至 16 中选择）
  设置传输数量：（从 1 至 65536 中选择）
  （有关块数与传输数量之间的区别，参见 3DMAC 的功能及操作。）

- 可指定是否发出传输成功完成和传输失败通知。
- 可选择以下五种传输模式：

  软件-块传输
  软件-突发传输
  硬件-命令传输
  硬件-块传输
  硬件-突发传输

传输模式

软件传输是通过 CPU 直接指令启动 DMAC 的方法。

硬件传输是外设发出传输请求时将外设的中断信号作为 DMAC 传输请求信号以直接启动 DMAC 的方法。

发送/接收数据或 A/D 转换数据需要传输时，多功能串行单元和 ADC 直接指示 DMAC 启动数据传输。外部中断单元和基本计时器单元直接指示 DMAC 在传输计时时启动数据传输。在这些情况下，数据可以通过预先设置转移来实现不用 CPU 传输。

缩写

本章包含下列术语：DE、DS、DH、PR、EB、PB、ST、IS、BC、TC、MS、TW、FS、FD、RC、RS、RD、EI、CI、SS、EM。这些术语参考 DMAC 控制寄存器（DMACR、DMACSA、DMACDA、DMACA、DMACB）的每位。参见 "5DMAC 寄存器"。

2. DMAC 配置

本节说明 DMAC 的系统配置及 I/O 信号。

2.1 DMAC 和系统配置
2.2 DMAC 的 I/O 信号

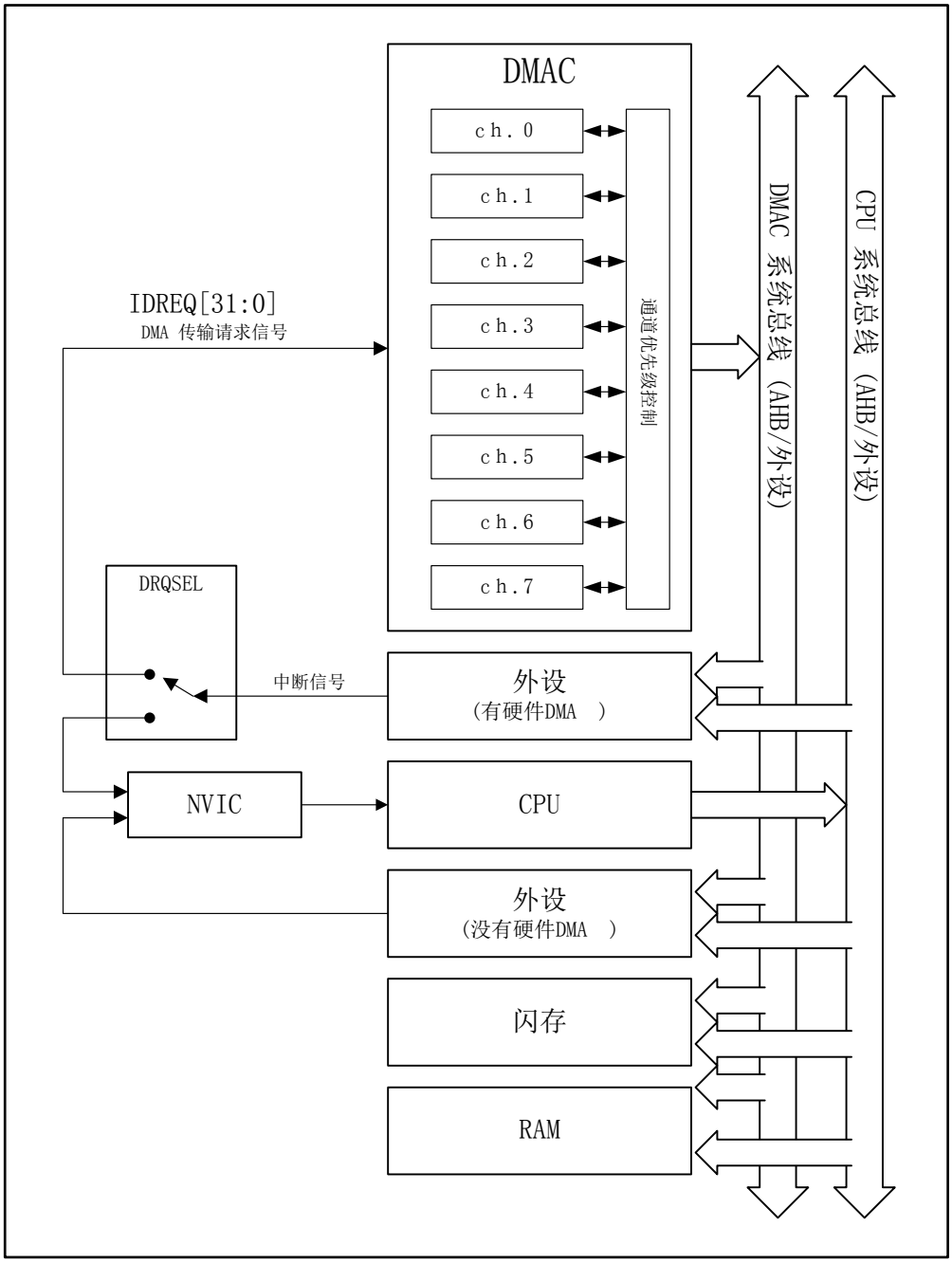
2.1 DMAC 和系统配置

本节说明 DMAC 及其系统配置

框图

Figure 2-1 所示为 DMAC 及其系统配置框图。

Figure 2-1 DMAC 及系统配置框图



框图说明

■ DMAC

DMAC 最多配置 8 条通道。各通道执行独立传输。通道之间发生冲突时，优先级控制器控制这些通道的传输操作。

■ 系统连接

为了方便说明，系统配置图在上图中已经简化了。更多详细信息，参见“系统概述”一章。 DMAC 通过系统总线连接至 CPU、闪存、RAM 及外设。DMAC 自身有独立于 CPU 总线的总线，可在访问 CPU 总线时执行传输操作。通过指定各通道传输目标地址和传输源地址以进行存储器与外设之间传输数据，系统中的任何地址区域均可访问。由于部分区域不能从 DMAC 访问，需检查内存映射。

■ 硬件传输请求信号的连接

被中断控制器（指 Figure 2-1 中的 DRQSEL）所选择支持硬件传输的外设中断信号，可以作为发送至 CPU 的中断信号或者作为发送至 DMAC 的 DMA 传输请求信号。

根据硬件请求执行 DMA 传输时，通过设置 DRQSEL，连接各外设的中断信号，作为提前发给 DMAC 的传输请求信号。不支持硬件传输的外设中断信号不能作为 DMA 传输请求信号。中断信号作为 DMAC 的传输请求信号时，不能作为 CPU 的中断信号。参见“中断”一章。

输入 DMAC 的 DMA 传输请求信号有 32 个。有关各信号与外设之间的对应关系，参见下一节中的 Table 2-1。

不能选择未集成外设的中断信号。应注意如果是多通道及多中断因素的外设，有些中断支持 DMA 传输，而其他则不支持。

硬件传输时，DMAC 各通道从上述 32 条传输请求信号中选择一条传输请求信号执行操作。使用 IS 寄存器进行选择。

■ 硬件传输请求清除信号的连接

支持硬件传输的一些外设需要在传输完成后清除传输请求信号（中断信号）。尽管 Figure 2-1 中未说明，此类外设的传输请求信号也要由 DRQSEL 选择并通过 DMAC 并将其清除。

■ 硬件传输停止请求信号的连接

多功能串行单元（以下简称 "MFS"）输出 DMA 传输停止请求信号。尽管 Figure 2-1 未作说明，DRQSEL 选择 MFS 时，MFS 传输停止请求信号连接至 DMAC。发出传输停止请求信号时，DMAC 停止传输操作。该配置用于屏蔽后续传输请求信号。

MFS 的传输停止请求信号生效条件如下：

- 如果接收到的中断被使能 (SCR:RIE=1)，则产生接收到的中断（SSR:PE 位、FRE 位或 ORE 位设置为 1）。
- 如果芯片选择的错误中断被使能 (SACSR:CSEIE=1)，则发生芯片选择错误中断（(SACSR:CSE 位设置为 1）。

■ DMAC 中断信号

尽管 Figure 2-1 未作说明，用于传输完成通知的中断信号连接至 NVIC。各通道有 8 个中断输出。

2.2 DMAC 的 I/O 信号

本节说明 DMAC 的 I/O 信号。

输入 DMAC 的传输请求信号

Table 2-1 列出输入 DMAC 的传输请求信号及对应外设的中断信号。

Table 2-1 传输请求信号和对应外设的中断信号列表

IDREQ 编号	对应外设的中断信号
0	保留
1	保留
2	保留
3	保留
4	保留
5	A/D 转换器单元 0 的扫描转换中断信号
6	A/D 转换器单元 1 的扫描转换中断信号
7	A/D 转换器单元 2 的扫描转换中断信号
8	基本计时器 ch.0 的 IRQ0 的中断信号
9	基本计时器 ch.2 的 IRQ0 的中断信号
10	基本计时器 ch.4 的 IRQ0 的中断信号
11	基本计时器 ch.6 的 IRQ0 的中断信号
12	MFS ch.0 的接收中断信号
13	MFS ch.0 的发送中断信号
14	MFS ch.1 的接收中断信号
15	MFS ch.1 的发送中断信号
16	MFS ch.2 的接收中断信号
17	MFS ch.2 的发送中断信号
18	MFS ch.3 的接收中断信号
19	MFS ch.3 的发送中断信号
20	MFS ch.4 的接收中断信号
21	MFS ch.4 的发送中断信号
22	MFS ch.5 的接收中断信号
23	MFS ch.5 的发送中断信号
24	MFS ch.6 的接收中断信号
25	MFS ch.6 的发送中断信号
26	MFS ch.7 的接收中断信号
27	MFS ch.7 的发送中断信号
28	外部中断单元 ch.0 的中断信号
29	外部中断单元 ch.1 的中断信号
30	外部中断单元 ch.2 的中断信号
31	外部中断单元 ch.3 的中断信号

DMAC 输出的中断信号

Table 2-2 列出 DMAC 输出的中断信号。

Table 2-2 DMAC 的中断信号列表

中断信号名称	中断因素寄存器	中断使能寄存器	中断类型
DIRQ0	DMACB0:SS[2:0]	DMACB0.CI	ch.0 传输成功中断
		DMACB0.EI	ch.0 传输失败中断
DIRQ1	DMACB1:SS[2:0]	DMACB1.CI	ch.1 传输成功中断
		DMACB1.EI	ch.1 传输失败中断
DIRQ2	DMACB2:SS[2:0]	DMACB2.CI	ch.2 传输成功中断
		DMACB2.EI	ch.2 传输失败中断
DIRQ3	DMACB3:SS[2:0]	DMACB3.CI	ch.3 传输成功中断
		DMACB3.EI	ch.3 传输失败中断
DIRQ4	DMACB4:SS[2:0]	DMACB4.CI	ch.4 传输成功中断
		DMACB4.EI	ch.4 传输失败中断
DIRQ5	DMACB5:SS[2:0]	DMACB5.CI	ch.5 传输成功中断
		DMACB5.EI	ch.5 传输失败中断
DIRQ6	DMACB6:SS[2:0]	DMACB6.CI	ch.6 传输成功中断
		DMACB6.EI	ch.6 传输失败中断
DIRQ7	DMACB7:SS[2:0]	DMACB7.CI	ch.7 传输成功中断
		DMACB7.EI	ch.7 传输失败中断

参考：中断产生因素及清除（有关详细信息，参见 "4DMAC 控制"。）

各通道的中断由以下因素产生：

- 通道传输成功后，将通道的 SS[2:0] 设置为 "101"。如通过 CI=1 将 SS[2:0] 设置为上述值（使能传输成功完成中断），则发生传输成功中断。
- 若通道传输失败，将通道的 SS[2:0] 设置为 "001"、"010"、"011" 及 "100"。如通过 EI=1 将 SS[2:0] 设置为上述值（使能传输失败中断），则发生传输失败中断。
- 传输成功中断和传输失败中断都需要执行逻辑 “**或**”，因此发生任一中断时，通道都将发出中断信号。

SS[2:0] 写入 "000"，清除各通道发出的中断信号。

3. DMAC 功能及操作

本节说明各传输模式下 DMAC 的操作。

3.1 软件-块传输
3.2 软件-突发传输
3.3 硬件-命令传输
3.4 硬件-块传输及突发传输
3.5 通道优先控制

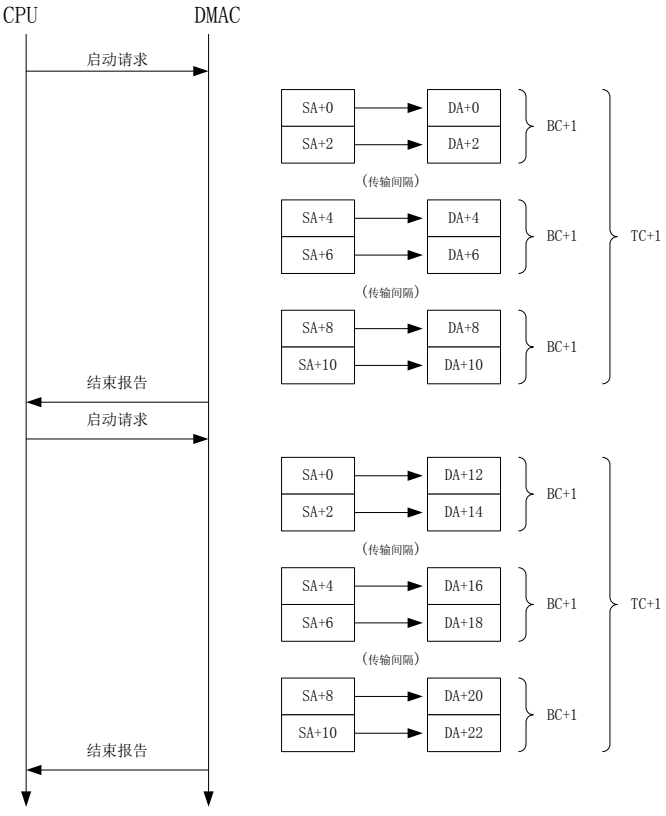
3.1 软件-块传输

本节说明软件-块传输。

Figure 3-1 所示为软件-块传输操作示例。示例中应用以下设置：

- 传输模式： 软件请求块传输 (ST=1, IS[5:0]=000000, MS=00)
- 传输源启动地址： SA(DMACSA=SA)
- 传输源地址控制： 可递增和重装 (FS=0, RS=1)
- 传输目标启动地址： DA(DMACDA=DA)
- 传输目标地址控制： 不可递增和重装 (FD=0, RD=0)
- 传输数据大小： 半字（16 位），块数= 2，
  传输数量= 3 (TW=01, BC=1, TC=2)
- BC/TC 重装： 可重装 (RC=1)

Figure 3-1 软件-块传输操作示例



通过 CPU 设置传输内容并指示启动传输后，DMAC 执行以下操作：

- 由于指定了传输数据位宽，传输以半字（16 位）执行。
- 根据传输源及传输目标的启动地址、数据位宽以及递增/固定的规定，在地址 SA 至地址 DA 区域执行块数 (=BC+1) 传输。
- 块传输时，每完成一次块传输则产生一个传输间隔。
- DMAC 根据传输数量 (=TC+1) 执行块数 (=BC+1) 数据传输。根据 CPU 传输请求所传输的数据量大小等于“数据宽 (TW) × 块数 (BC+1) × 传输数量 (TC+1)”。
- 传输完成后，DMAC 通知 CPU 传输完成。
- 如传输完成后再次指示启动传输，因传输源地址已设置为重装 (RS=1)，将从之前的传输启动地址 (SA+0) 重新启动传输。如果未将传输目标地址设置为重装 (RD=0)，将从先前传输结束地址的下一地址 (DA+12) 启动传输。同样，如果指定了 BC/TC 重装，下次传输将重装与上次传输相同的块数和传输数量值。

传输间隔期间不执行传输，插入传输间隔的目的是防止某一条 DMAC 通道占用系统总线的访问权。如多条通道均有传输请求，DMAC 切换在传输间隔计时期间执行传输操作的通道。通过调节 BC 及 TC 的设置可控制传输间隔生成的频率。

此外，传输间隔计时期间总线访问权也转至 CPU。此产品的系统总线为多层配置，带有专用于 DMA 的特殊系统总线。因此，如 CPU 与访问目标之间无冲突，CPU 操作的同时可执行传输。即使 CPU 与访问目标之间发生冲突，只要 DMAC 传输处在不同的地址域群（RAM 和外设，或闪存和 RAM 等），CPU 操作几乎不受影响。但如传输处在相同的地址域群（RAM 和 RAM 等），CPU 操作和/或系统性能可能会受影响，具体取决于所用块数，对此必须引起注意。

（上述“地址域群”指通过相同总线桥连接在 AHB 系统总线上的地址区域群。）

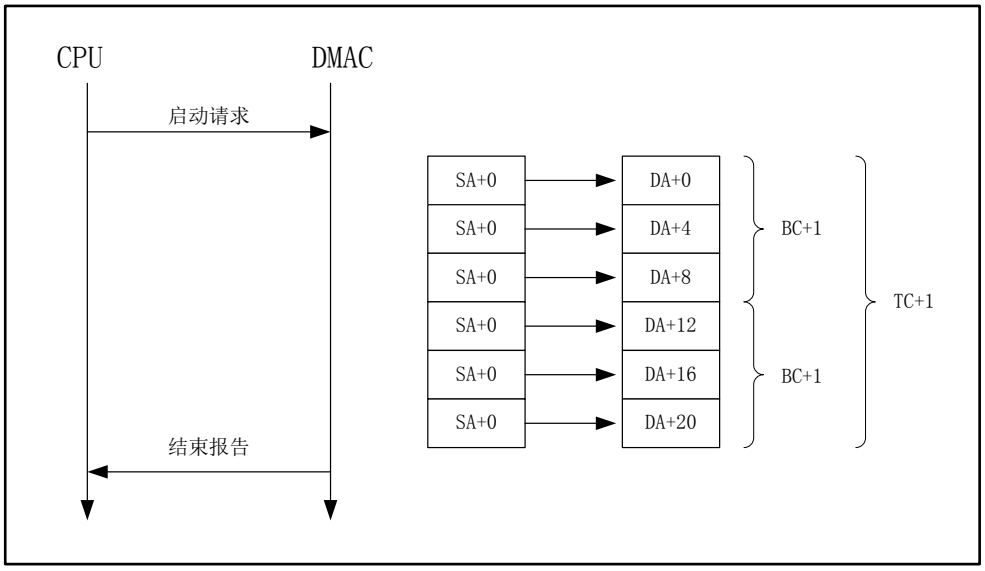
3.2 软件-突发传输

本节说明软件-突发传输。

Figure 3-2 所示为软件-突发传输操作示例。示例中应用以下设置：

- 传输模式： 软件请求突发传输 (ST=1, IS[5:0]=000000,MS=01)
- 传输源启动地址： SA(DMACSA=SA)
- 传输源地址： 可固定和重装 (FS=1, RS=1)
- 传输目标启动地址： DA(DMACDA=DA)
- 传输目标地址： 不可递增和重装 (FD=0, RD=0)
- 传输数据大小： 字（32 位），块数=3，传输数量=2(TW=10,BC=2,TC=1)
- 传输数量重装： 重装的传输数量 (RC=1)

Figure 3-2 软件-突发传输操作示例



通过 CPU 设置传输内容并指示启动传输后，DMAC 执行以下操作：

- 由于指定了传输数据位宽，传输以字（32 位）执行。
- 根据传输源及传输目标的启动地址、数据位宽以及递增/固定的规定，在地址 SA 至地址 DA 区域执行块数 (=BC+1) 传输。由于指定了固定的传输源地址，传输源地址与传输源启动地址 (SA+0) 相同。
- 突发传输时，连续执行传输，不产生传输间隔。
- DMAC 根据传输数量 (=TC+1) 执行块数 (=BC+1) 数据传输。根据 CPU 传输请求所传输的数据大小等于"数据位宽 (TW) × 块数 (BC+1) × 传输数量 (TC+1)。"
- 传输完成后，DMAC 通知 CPU 传输完成。

与块传输不同的是突发传输不产生传输间隔。由于受控通道占用了系统总线访问权，该通道可作为优先通道。

3.3 硬件-命令传输

本节说明硬件-命令传输。

根据 MFS 和 ADC 的外设传输请求信号执行 DMA 传输时，使用硬件-命令传输。

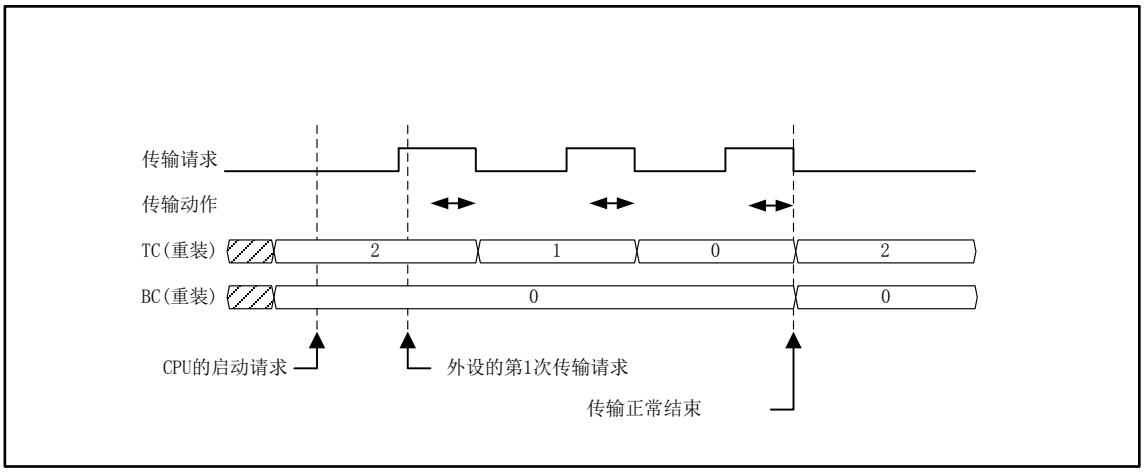
硬件-命令传输方法用于根据信号电平接收外设的传输请求信号。如传输请求信号为高电平，则执行传输。如传输请求信号为低电平，则不执行传输。执行传输时，如果存在传输数据，则将各外设输出的中断信号设置为高电平（带有中断请求），或者没有传输数据时，将中断信号设置为低电平（无中断请求）。

使用硬件-命令传输时，块数始终指定为 1 (BC=0) 。

Figure 3-3 所示为硬件-命令传输操作示例。示例中应用以下设置。其中省略了传输源及传输目标的地址设置以及传输数据位宽的设置。

- 传输模式： 硬件-命令传输（ST=0，IS= 传输请求源的外设，MS=10）
- 传输数据大小： 块数= 1，传输数量= 3 (BC=0,TC=2)

Figure 3-3 软件命令传输操作示例



硬件-命令传输操作如下所述：

通过 CPU 指定传输内容，指令启动操作。 DMAC 等待外设的传输请求。收到传输请求后，DMAC 执行一次传输，然后等待下一次传输请求。等待时产生传输间隔。每次收到传输请求时，DMAC 执行传输数量 (TC+1) 的相同操作。执行的传输总数为 (TC+1)。将外设的传输请求与 DMAC 传输数量 (TC+1) 相匹配。完成所有传输后，DMAC 通知 CPU 传输完成。

3.4 硬件-块传输及突发传输

本节说明硬件-块传输及突发传输。

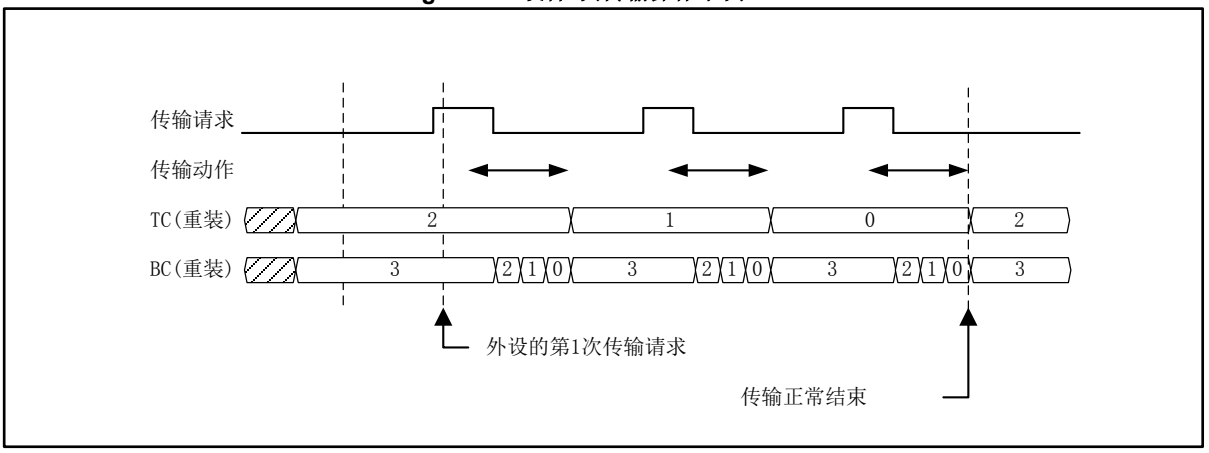
根据基本计时器外设的传输请求信号或外部中断执行 DMA 传输时，要使用到硬件-块传输或硬件-突发传输。

硬件-块传输及硬件-突发传输用于接收处于上升沿的传输请求信号。检测到传输请求信号的上升沿时执行传输。通过外设输出的中断信号指定 DMAC 传输启动计时。

Figure 3-4 所示为硬件-块传输操作示例。示例中应用以下设置。其中省略了传输源及传输目标的地址设置以及传输数据位宽的设置。

- 传输模式： 硬件-块传输（ST=0，IS= 传输请求源的外设，MS=00）
- 传输数据大小： 块数= 4，传输数量= 3 (BC=3,TC=2)

Figure 3-4 硬件-块传输操作示例



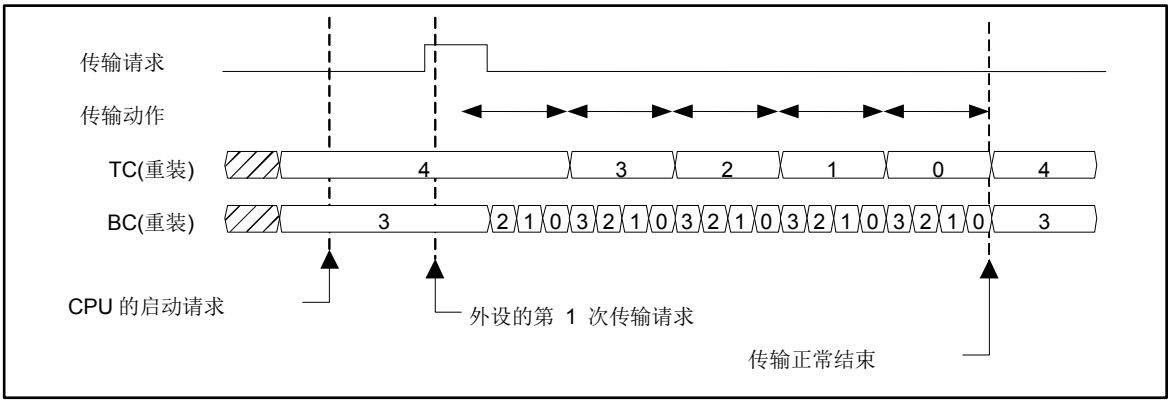
硬件-块传输操作如下：

通过 CPU 指定传输内容，指令启动操作。 DMAC 等待外设的传输请求。收到传输请求后，DMAC 执行块数 (=BC+1) 传输，然后等待下一次传输请求。等待时产生传输间隔。每次收到传输请求时，DMAC 执行传输数量 (TC+1) 的相同操作。执行的传输总数为 (BC+1) × (TC+1)。将外设发出的传输请求与 DMAC 传输数量 (TC+1) 相匹配。完成所有传输后，DMAC 通知 CPU 传输完成。

Figure 3-5 所示为硬件-突发传输操作示例。示例中应用以下设置。其中省略了传输源及传输目标的地址设置以及传输数据位宽的设置。

- 传输模式：　　输请求源的外设，MS=01）
- 传输数据大小：　块数= 4，传输数量= 5 (BC=3,TC=4)

Figure 3-5 硬件-突发传输操作示例



硬件-突发传输操作如下所述：

通过 CPU 指定传输内容，指令启动操作。 DMAC 等待外设的传输请求。收到第一次传输请求后，DMAC 执行通过 (BC+1)×(TC+1) 计算出的所有次数的传输。硬件-突发传输过程不产生传输间隔。完成所有传输后，DMAC 通知 CPU 传输完成。

3.5 通道优先控制

本节说明通道优先控制。

通道优先控制

如多通道均有传输请求，各通道传输间隔计时期间 DMAC 切换其中要传输的通道。此时根据优先控制确定执行传输的下一条通道。可通过 PR 从固定优先或循环优先中选择优先控制。Figure 3-6 所示为说明图。此图中 X 轴表示时间轴。箭头表示所有通道同时发出传输请求时各通道执行其传输操作的传输计时。

固定优先模式操作 (PR=0)

固定优先模式下，所有接到传输请求的通道中最小编号通道优先执行传输操作。

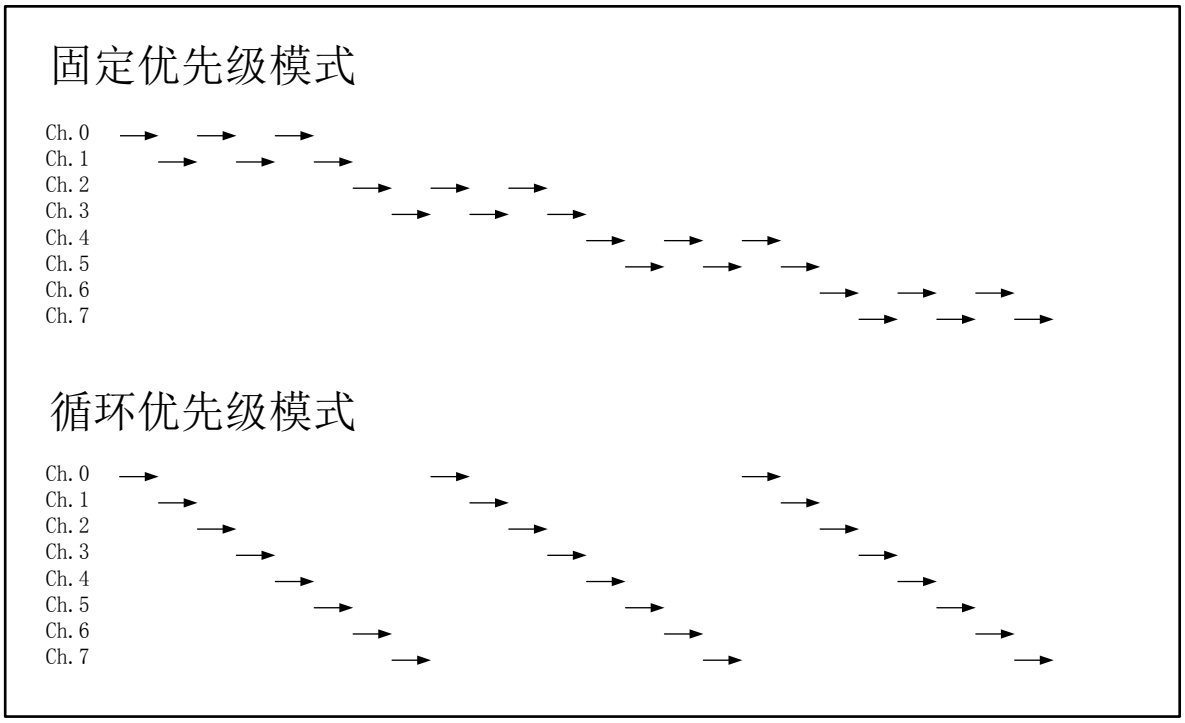
（优先顺序：ch.0>ch.1 > ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7）

首先执行最高优先级通道的传输（Figure 3-6 中的 ch.0）。最高优先级通道在传输间隔计时期间停止传输操作时，执行第二优先通道的传输操作（Figure 3-6 中的 ch.1）。因此，第一优先通道和第二优先通道交替执行传输操作。此后，前一通道完成传输时，后一通道启动传输操作（Figure 3-6 中的 ch.3）。

循环优先模式操作 (PR=1)

循环优先模式下，所有通道均等地执行其传输操作。

Figure 3-6 通道优先控制说明图



4. DMAC 控制

本节详细地说明 DMAC 控制方法。

4.1 DMAC 控制概述
4.2 软件传输的 DMAC 操作及控制步骤
4.3 硬件 (EM=0) 传输的 DMAC 操作及控制步骤
4.4 硬件 (EM=1) 传输的 DMAC 操作及控制步骤

4.1 DMAC 控制概述

本节概述 DMAC 控制。

DMAC 各通道的控制寄存器均有 EB（单通道操作使能位）和 PB（单通道停止位）。通道可通过这些位的操作控制 DMA 传输操作的启动（使能操作）、强行终止（停止操作）及暂停。控制寄存器也配有 DE（全部通道操作使能位）和 DH（全部通道暂停位），可同时控制所有通道的传输操作。

各通道原为禁止操作状态（禁用状态），此状态下各通道的传输内容（传输源及传输目标的地址、传输数据宽、传输数量、传输模式等）在其配置寄存器中指定。然后通过写入 EB、PB、DE 及 DH 控制传输操作，指令操作启动或暂停。

各通道完成传输后将结束码设置为 SS（停止状态），通知其停止状态。传输完成后可生成中断。传输完成后，各通道清除 EB 和 PB 并返回至禁止操作状态（禁用状态）。

以下章节描述根据软件请求进行 DMA 传输以及根据外设的传输请求进行硬件 DMA 传输的操作和控制步骤。

下列术语用于解释 CPU 的指令，参考以下值写入 EB、PB、DE 及 DH 位：

- 使能单通道操作指令（写入 EB=1，PB=0）
- 禁用单通道操作指令（写入 EB=0）
- 暂停单通道操作指令（写入 EB=1，PB=1）
- 使能全部通道操作指令（写入 DE=1，DH=0000）
- 禁用全部通道操作指令（写入 DE=0）
- 暂停全部通道操作指令（写入 DE=1，DH!=0000）

4.2 软件传输的 DMAC 操作及控制步骤

本节说明软件传输的 DMAC 操作及控制步骤。

Figure 4-1 软件 DMA 传输状态转换图

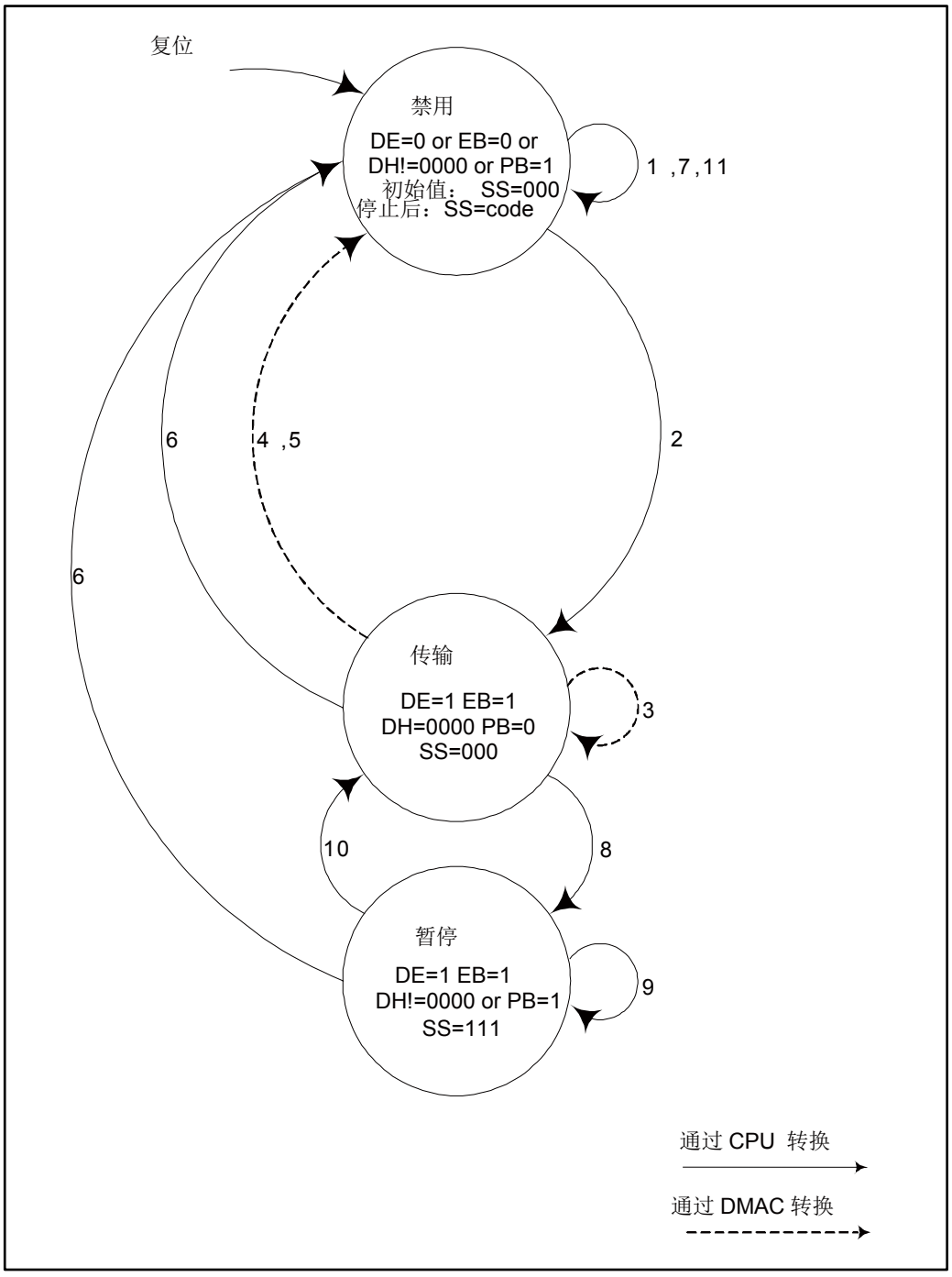


Figure 4-1 所示为软件传输时所控制的通道状态转换图。图中转换线旁边的数字与下述控制步骤中出现的数字相对应。转换实线表示 CPU 指令的状态转换，而虚线表示 DMAC 操作引起的状态转换。

各状态描述

■ 禁用状态

此状态下禁止通道传输。此状态下通道不进行操作，等待 CPU 发出的指令。系统复位时，禁用状态适用 DE=0、EB=0、DH=0000 和 PB=0。

■ 传输状态

此状态下使能受控通道传输。此状态下通道执行指定的传输操作。所有传输操作完成后，通道返回到禁用状态。状态也会按 CPU 指令改变。

■ 暂停状态

此状态下受控通道根据 CPU 发出的暂停指令暂停传输操作，等待 CPU 发出其他指令。

控制步骤说明

1. 禁用状态/传输准备

   通过 CPU 指定受控通道的传输内容（写入 DMACSA、DMACDA、DMACA 及 DMACB）。有关指定传输内容的详细信息，参见“5. DMAC 寄存器”。完成传输后 DMAC 生成中断，此时设置 EI 及 CI。

   下列限制条件适用于软件传输。指定 ST=1 且 IS [5:0] =000000。命令传输模式不能指定为 MS。始终将 EM 设置为 "0"。

   发出指令使能全部通道操作并设置 PR。进行步骤 2 时可同时将数据写入 DMACA。

2. 禁用状态 =>传输状态/传输启动

   CPU 发出使能单通道操作指令。设置 DE=1、EB=1、DH=0000 且 PB=0 时，受控通道转为传输状态。

3. 传输状态

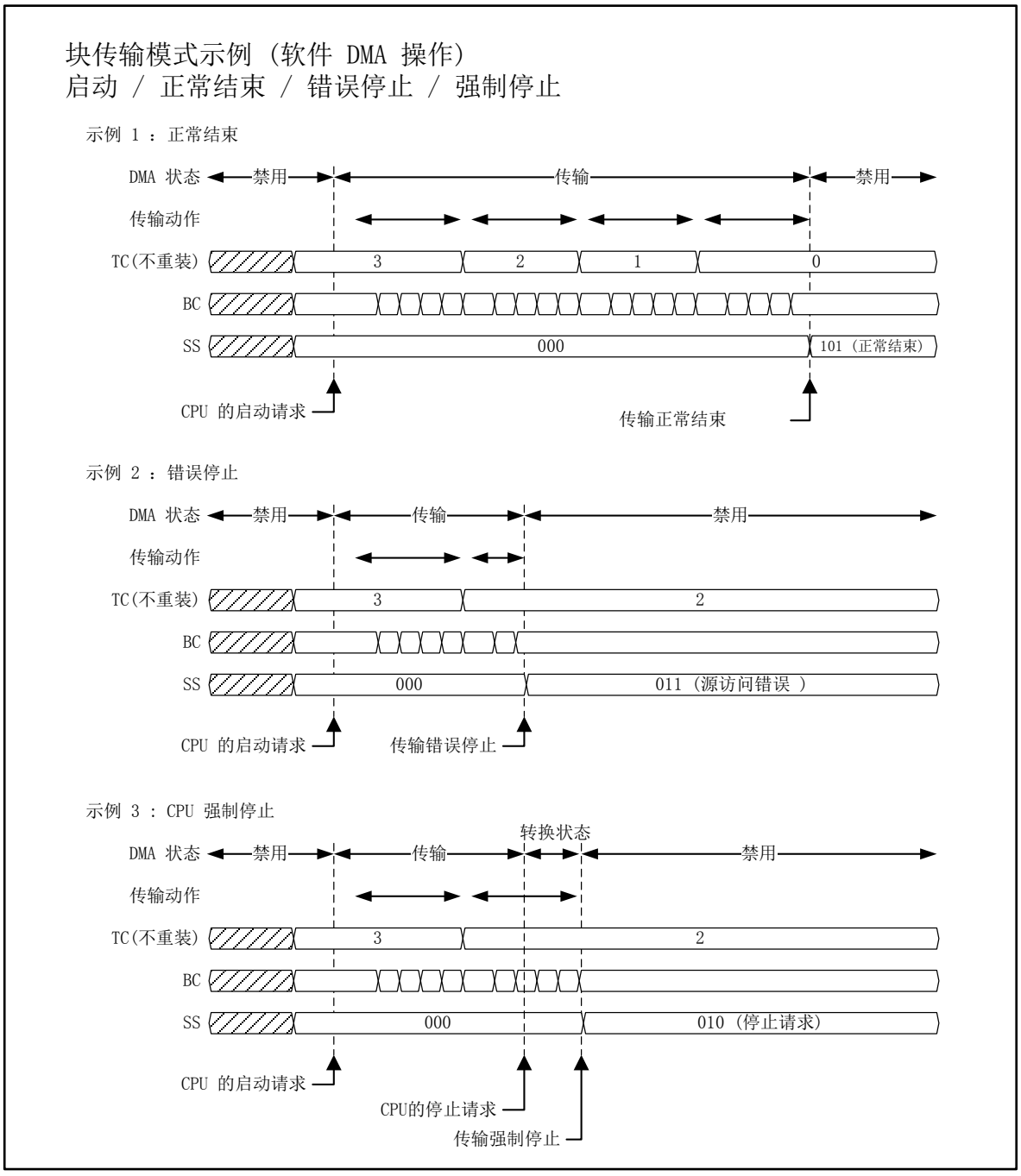
   当传输状态的通道被启用访问系统总线时，该通道根据传输内容执行传输（传输启动时间取决于其他通道的状态）。块传输模式下，每次更新 TC 时生成传输间隔。突发传输模式下不生成传输间隔。传输操作时，BC、TC、DMACSA 及 DMACDA 表示剩余传输数量及该时间点的传输地址。通过读取 CPU，可检查传输状态。

   不能通过 CPU 将指定的传输内容改变至传输状态下的通道（重写 DMACSA、DMACDA、DMACA[29:0]、DMACB[31:1]）。（但可重写 EB、PB 及 EM。）

4. 传输状态=>禁用状态/传输成功完成

   根据 (BC+1)×(TC+1) 计算出的传输次数都成功完成后，传输状态下的通道清除 EB、PB 及 ST 并转为禁用状态。通道设置 SS=101，以通知传输成功。参见 Figure 4-2 中的示例 1。如 CI 使能传输成功中断，则发生中断。如指定 BC、TC、DMACSA 及 DMACDA 重装，则根据指定的传输内容执行重装。

Figure 4-2 软件-块传输操作示例



5. 传输状态 =>禁用状态/传输错误停止

   传输状态的通道暂停传输进程，如地址溢出，则出现传输源访问错误或传输目标访问错误。清除 EB、PB 及 ST 并转为禁用状态。设置指示错误内容的值至 SS[2:0]，发出错误停止通知。参见 Figure 4-2 中的示例 2。如 EI 使能传输失败中断，则发生中断。未指定重装的 BC、TC、DMACSA 及 DMACDA 保持传输暂停时设置的值。

尝试访问系统总线中不存在的地址域或访问禁止 DMAC 访问的地址域时，通常会出现传输错误。一般应用不会出现这种错误。

6. 传输状态、暂停状态 =>禁用状态/强行停止传输

   从 CPU 向传输或暂停状态下的通道发出禁用单个通道操作或禁用全部通道操作的指令时，该通道的传输操作被强行停止（有关向禁用状态下的通道发出禁用操作指令时的操作，参见软件控制的步骤 11）。

   如 CPU 发出指令，则相关通道暂停其传输进程。清除 EB、PB 及 ST 并转为禁用状态。设置 SS[2:0]=010，并通知该通道的传输被强行停止。如 EI 使能传输失败中断，则发生中断。未指定重装的 BC、TC、DMACSA 及 DMACDA 保持传输暂停时设置的值。

   如 Figure 4-2 中示例 3 所示，CPU 发出指令后，相关通道暂停执行传输计时期间（传输启动前的传输间隔时间内）传输停止。若通道处于暂停状态，传输立即停止。指令与停止之间有时间差（转换状态）。所需时间取决于 BC 设置。由于此期间不能设置或启动新的传输，确保下一次传输设置前已停止操作。

   如指令为禁用全部通道操作，则停止的时序视通道的不同而变化。由于所有通道停止时设置了 DS，可确认所有通道均已停止。

   因传输模式（突发/块/命令）及传输状态（执行的传输数量以及禁用操作指令的计时）等因素，即便 CPU 发出指令，也可能不会被强行停止传输，而是成功完成。同样，如传输停止前发生传输错误，则错误停止适用于该传输。

7. 禁用状态/传输后的进程

   CPU 读取 SS，检查传输的完成状态。CPU 清除 SS，准备下次传输。如果已使能中断，清除 SS 可使 DMAC 的中断信号失效。

   若传输成功，CPU 将按要求复位传输内容。如每次均指定了重装，那么在传输启动前应将值重装到 BC, TC, DMACSA 及 DMACDA 中。如未指定每次重装，则 BC 及 TC 被初始化为 0。DMACSA 及 DMACDA 显示下一传输地址。

   错误停止和强行停止时，由于 BC、TC、DMACSA 及 DMACDA 的值可能为暂停时设置的值，所以必须将其复位。

   如收到禁用全部通道操作的指令后停止传输，DE 设置为 0，因此下一个传输需要使能全部通道操作以及使能单通道操作的指令。

8. 传输状态/传输暂停

   CPU 向传输状态下的通道发出暂停单通道操作或暂停全部通道操作指令时，该通道的传输操作暂停（有关向禁用状态下的通道发出暂停操作指令时的操作，参见软件控制的步骤 11）。

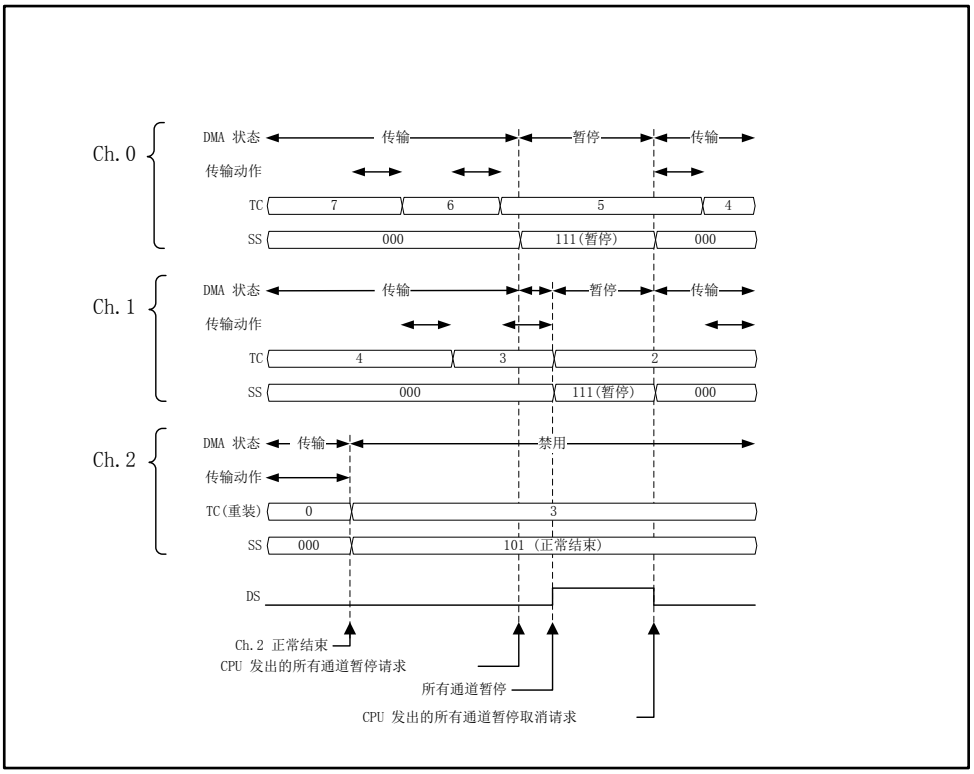
   CPU 发出指令时，相关通道暂停传输进程。设置 SS=111 并发出暂停状态的通知。此时无中断生成。

   CPU 发出指令后，相关通道暂停执行传输计时期间（传输启动前的传输间隔期内）传输停止。指令与停止之间有时间差（转换状态）。所需时间取决于 BC 设置。参见 Figure 4-3。

   收到暂停全部通道操作指令时，停止的时序视通道的不同而变化。由于所有通道停止时设置了 DS，可确认所有通道均已停止。参见 Figure 4-3。

因传输模式（突发/块/命令）及传输状态（执行的传输数量和暂停操作指令计时）等因素，即便 CPU 发出指令，传输可能不会被暂停，而是成功完成。同样，如传输停止前发生传输错误，则该传输因错误停止。

Figure 4-3 收到暂停全部通道指令时的操作



9. 暂停状态

从 CPU 中读取 SS，以确定传输已暂停。暂停状态下通道的 SS 为 111，而此状态下不能从 CPU 中将其清除。

即便暂停时也不能指定或变更传输内容（重写 DMACSA、DMACDA、DMACA[29:0] 或 DMACB[31:1]）。同样，暂停状态下的通道收到暂停指令后将继续保持暂停状态。

10. 暂停状态/传输暂停的取消

如果通道受单通道操作暂停指令处于暂停状态，则向此通道发出单通道操作使能指令后，该通道返回至传输状态。如果通道受全部通道操作暂停指令处于暂停状态，则向这些通道发出全部通道操作使能指令后，这些通道返回至传输状态。如发出了两种暂停指令，再发出一个指令可取消这两种指令。

收到指令后，通过 DMAC 将 SS[2:0] 清除为 000。

暂停状态下发出使能单通道操作指令和使能全部通道操作指令时，是指令取消暂停。禁用状态下发出上述指令时，要注意可能是指令启动新的传输。参见软件控制的步骤 11。

Figure 4-3 所示为暂停全部通道操作指令示例。该图的说明如下所述：

块传输模式时，开始有三条通道（即 ch.0、ch.1 及 ch.2）执行传输操作。ch.2 成功完成传输后变为禁用状态并设置 SS[2:0]=101。然后 ch.0 和 ch.1 交替执行传输。

如此时 CPU 发出暂停全部通道操作指令，则适用以下操作：因受传输间隔计时的限制， ch.0 立即变为暂停状态并设置 SS[2:0]=111。ch.1 处于传输操作的中间，执行传输操作持续直至下一传输间隔的计时，然后转为暂停状态并设置 SS[2:0]=111。ch.2 处于禁用状态，将保持禁用状态且不改变 SS。所有通道停止操作时设置 DS。

接下来，如果 CPU 发出使能全部通道操作指令（取消暂停指令），则适用以下操作。ch.0 及 ch.1 返回至传输状态并清除 SS[2:0] 为 "000"。因 ch.2 处于禁用状态 (DE=1, EB=0)，保持此状态不启动操作。因取消了所有通道的暂停，DS 复位。

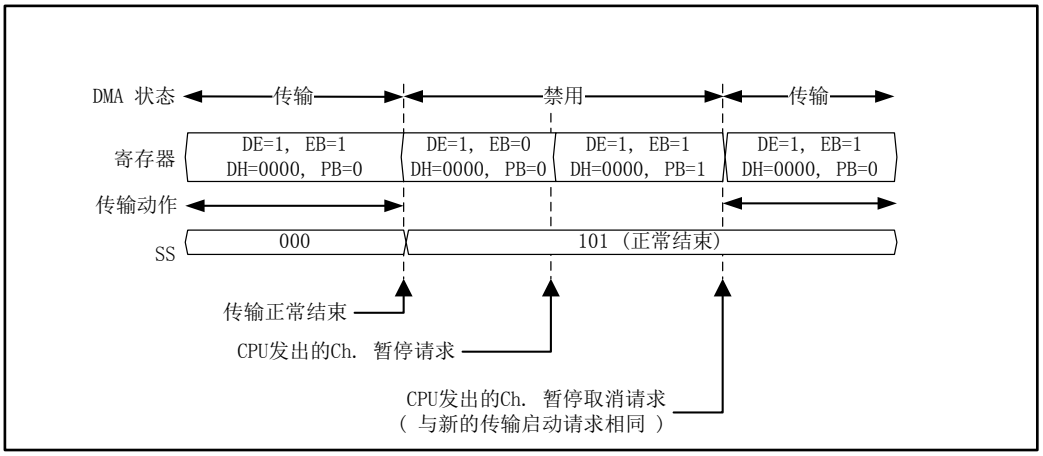
11. 禁用状态的操作

除非设置 DE=1、DH=0000、EB=1 且 PB=0，否则禁用状态下的通道将保持禁用状态。即使是软件控制步骤的 1-2 中，在 DE=0 且 EB=0 条件时设置 DE，然后设置 EB。也可以在设置 DE 前先设置 EB。可在用于传输的多通道传输设置全部完成后，最后设置 DE。此时，可发出指令同时启动多通道传输操作。如果发出此类同时启动传输指令，DMAC 将根据 PR 设置（只有禁用全部通道操作时才可设置或改变 PR）选择启动传输的通道。

如向禁用状态的通道发出禁用单通道操作指令、暂停单通道操作指令、禁用全部通道操作指令或暂停全部通道操作指令，仅改变 DE、DH、EB 及 PB 设置，而不会设置 DE=1、DH=0000、EB=1 和 PB=0。因此相关通道暂停操作，也不会改变 SS[2:0]。如 Figure 4-3 中 ch.2 操作示例所示，若 CPU 向禁用状态下的通道发出暂停全部通道操作指令，该通道不会改变状态，SS[2:0] 指示前一个传输的完成。

如向禁用状态的通道发出暂停单通道或全部通道操作指令，该通道可能通过 DE=1、EB=1 (DH!=0000 或 PB=1)被设为禁用状态。尽管此状态的位值与 DE、EB、DH 及 PB 相同，由于 SS[2:0] 的值不同，也可对其进行区分。（如 Figure 4-4 所示。）

Figure 4-4 禁用状态下发出暂停单通道操作指令的操作示例



某一通道执行传输操作。 CPU 向该通道发出暂停单通道操作指令。传输完成后发出该指令，通道转为禁用状态（DE=1、DH=0000、EB=0、PB=0）。出现此现象的原因是正执行传输操作的通道改变为 CPU 指令之外的状态。此时，相关通道位的值根据 CPU 的指令变为（DE=1、DH=0000、EB=1、PB=1），但 SS[2:0] 仍保持为 101（传输完成后设置的值）。如果受暂停指令操作停止，SS[2:0] 为 111，所以可能就有暂停状态和传输完成后状态的区分。应注意，如 Figure 4-4 所示，若未通过 SS[2:0] 检查通道状态就发出取消暂停指令，则会意外启动新的传输。

- 补充说明 1

由于 ST 在传输完成后清除，传输完成后 ST 的读取值为 0。如果是软件传输，要注意无论 ST 的读取值为何，ST 必须总是写入 1 。

- 补充说明 2

  前一个单通道操作使能指令指示启动传输后以及确认该传输完成之间期间，不能发出单通道操作使能指令。其原因是受控通道可能改变为 CPU 指令之外的状态，且 DMAC 转变为禁用状态 (EB=0) 时也可能发出启动新传输的指令。即便 SS[2:0] 值确认受控通道是传输状态，受控通道也可能在该时间点与写入操作之间的时间内转换为禁用状态。

- 补充说明 3

  只有通过 CPU 才能重写 DE 和 DH，不能从 DMAC 中清除这些寄存器。因此传输操作期间可以写为 DE=1 及 DH=0000。
  如果向全部通道暂停状态（DE=1、DH!=0000、EB=1、PB=0）的通道发出禁用单通道操作指令，不会清除 DH。收到指令后，相关通道变为禁用状态（DE=1、DH!=0000、EB=0、PB=0）。为了启动相关通道新的传输，写入 DE=1 及 DH=0000。这表示需要取消所有操作的暂停，才能启动新的单通道传输。

- 补充说明 4

  传输完成后通过 DMAC 设置 SS[2:0] 值，只要是禁用状态就不能通过 DMAC 重写 SS[2:0] 值。即便未清除 SS[2:0] 值也可启动下一次传输。但如果转为传输状态， SS[2:0] 值可以（或不可以）通过 DMAC 清除。使用 DMAC 的中断时，应注意，如果未清除 SS[2:0] 就转换为传输状态，会在 CPU 未指示的时序使中断信号失效。

4.3 硬件 (EM=0) 传输的 DMAC 操作及控制步骤

本节说明硬件 (EM=0) 传输的 DMAC 操作及控制步骤。

Figure 4-5 硬件 (EM=0) 传输状态转换图

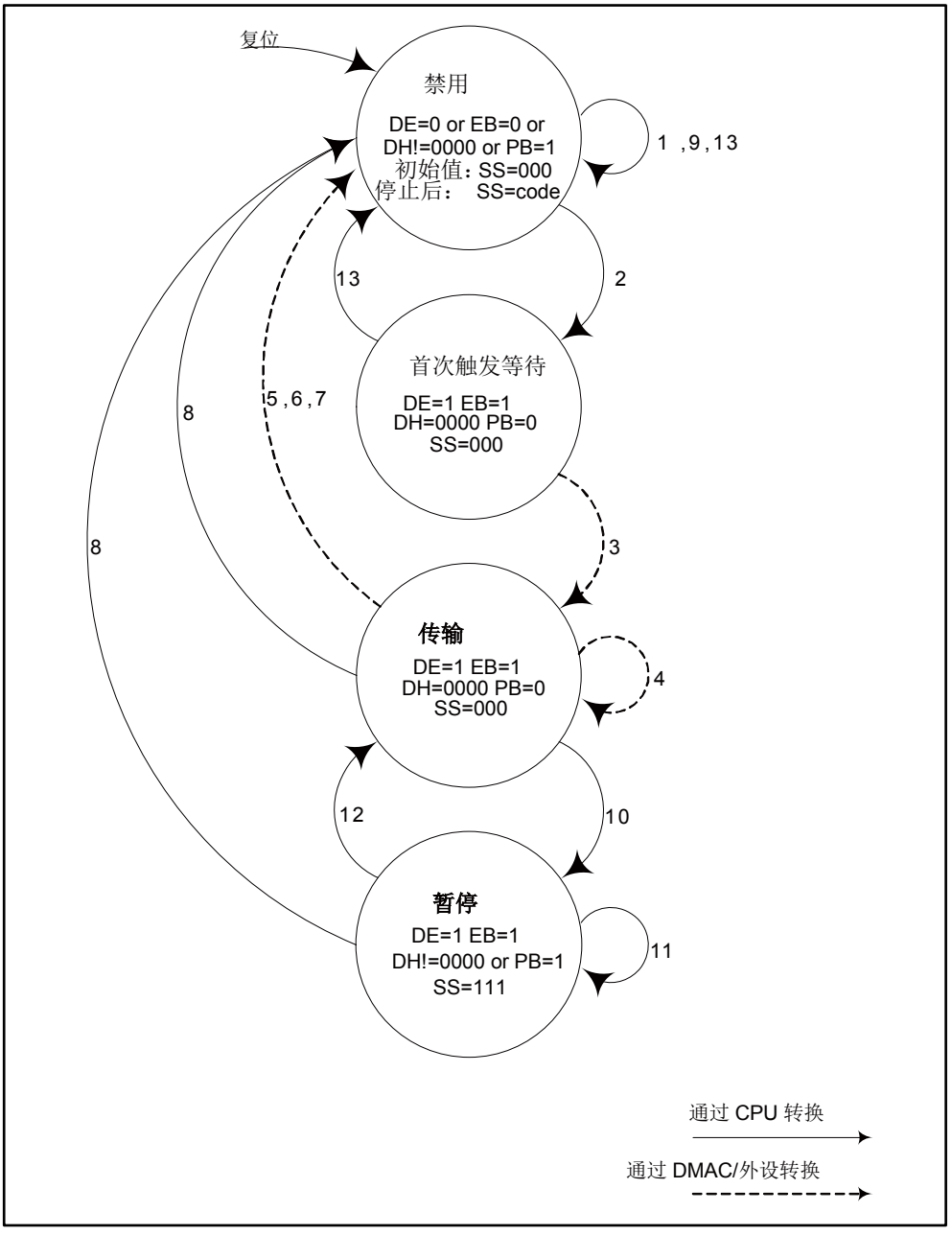


Figure 4-5 所示为硬件 (EM=0) 传输中受控通道的状态转换图。图中转换线旁边的数字与下述控制步骤中出现的数字相对应。转换实线表示 CPU 指令的状态转换，而虚线表示 DMAC/外设操作引起的状态转换。

以下说明中的“参见软件传输控制步骤” 软件传输控制步骤中相同的控制表示适用，无需特别说明，因此省略了冗赘解释。此说明示例中，假设设置 EM=0。

各状态描述

■ 禁用状态

参见 "4.2 软件传输的 DMAC 操作及控制步骤" 中的控制步骤说明。

■ 首次触发等待状态

此状态下，使能受控通道执行传输。此状态的通道等待外设发出首次传输请求。CPU 发出指令后通道也会改变其状态。

■ 传输状态

此状态时，受控通道已收到外设的首次传输请求。此状态下的通道执行指定的传输操作。所有传输操作完成后，通道返回禁用状态。CPU 发出指令后通道也会改变其状态。

■ 暂停状态

参见"4.2 软件传输的 DMAC 操作及控制步骤" 中的控制步骤说明。

控制控制步骤说明

1. 禁用状态/传输准备

   参见软件传输控制的步骤 1。
   下列限制条件适用于硬件传输。
   使用中断控制器模块（参见 “中断”一章中的“4.1 DMA 请求选择寄存器 (DRQSEL)”）预先确定外设中断信号，作为传输请求信号发送至 DMAC 。同时通过 IS 设置 ST=0，并指定传输执行通道需处理的外设传输请求。多条通道不能处理相同外设的传输请求。如果是命令传输模式，设置 BC=0。本节说明 EM=0 时的操作。

2. 禁用状态 =>首次触发等待状态/传输使能

   CPU 发出使能单通道操作的指令。设置 DE=1、EB=1、DH=0000 且 PB=0 时，受控通道转为首次触发等待状态。

3. 首次触发等待状态/传输启动

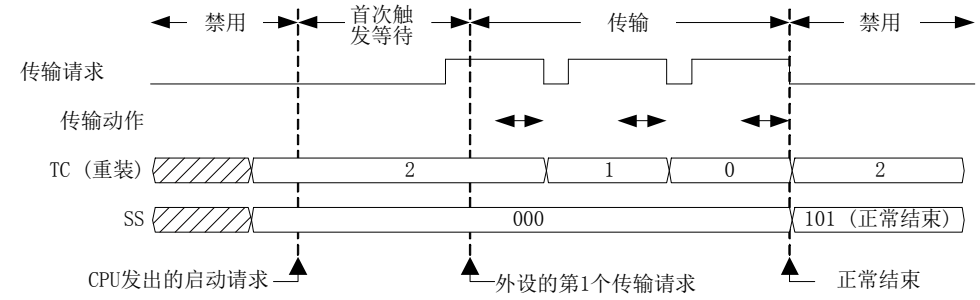
   首次触发等待状态下的通道等待外设发出传输请求信号或等待 CPU 的指令。发出首次传输请求信号后，通道变为传输状态。

4. 传输状态

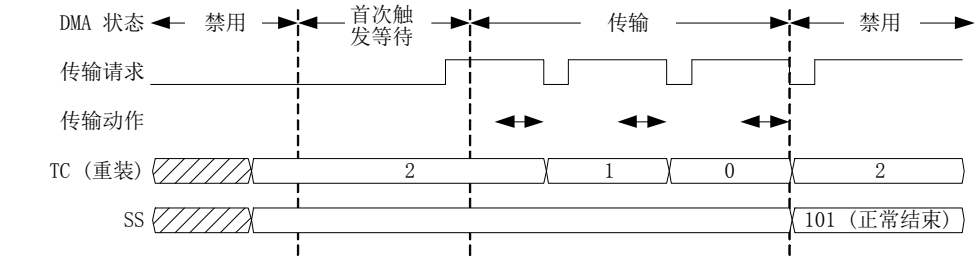
   参见软件传输控制的步骤 3。
   硬件传输时，如 “第 3.3 节硬件-命令传输” 和 “第 3.4 节硬件-块传输与突发传输” 所述，传输状态的通道根据外设的传输请求信号执行传输操作。各模式下，外设的传输请求数量匹配 DMAC 所要求的传输请求数量。下文说明各操作模式下传输请求数量多于或少于需求数量时的操作。
   Figure 4-6 所示为命令传输示例。命令传输时，需完成传输的传输请求数量为 TC+1。除非传输请求数量多于或少于请求数量，否则 CPU 不会介入（Figure 4-6 中的示例 1）。
   如外设生成的传输请求数量超过 DMAC 的设置传输数量，指定传输数量完成后 DMAC 变为禁用状态。禁用状态下不会执行其他传输。同样，由于未从 DMAC 中清除多余的传输请求信号，生效的状态将持续（Figure 4-6 中的示例 2）。
   如外设生成的传输请求数量少于 DMAC 的设置传输数量，DMAC 将保持传输状态并等待剩余数量的传输请求（Figure 4-6 中的示例 3）。
   DMAC 的传输进程可能会太慢，不能跟上外设传输请求的生成间隔。如果是命令传输，则保持传输请求信号的生效。因此可执行的传输数量达到 TC+1（Figure 4-6 中的示例 4）。

Figure 4-6 硬件-命令传输操作

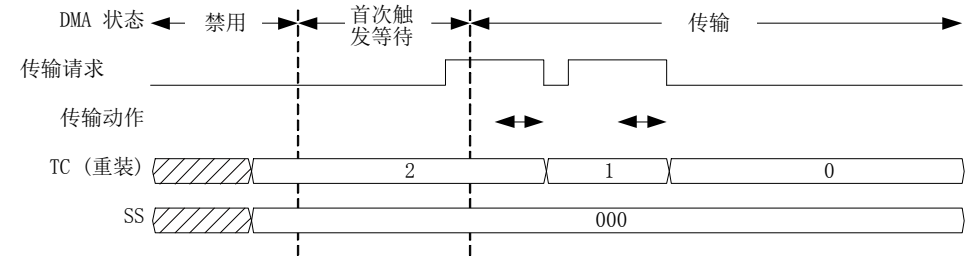
示例 1 ： (TC+1)== 外设发出的传输请求



示例 2: (TC+1) < 外设的传输请求



示例 3: (TC+1) > 外设的传输请求



示例 4: 延迟 DMA 传输

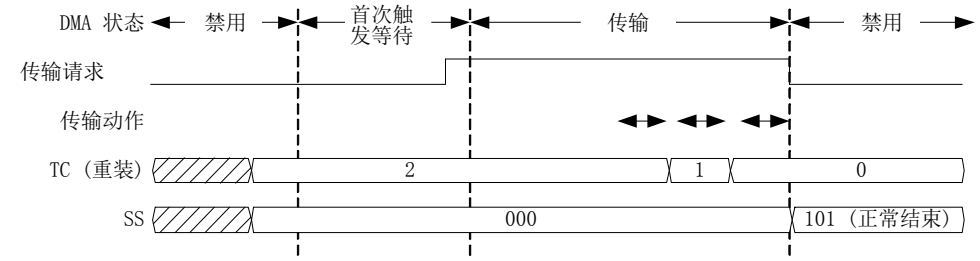
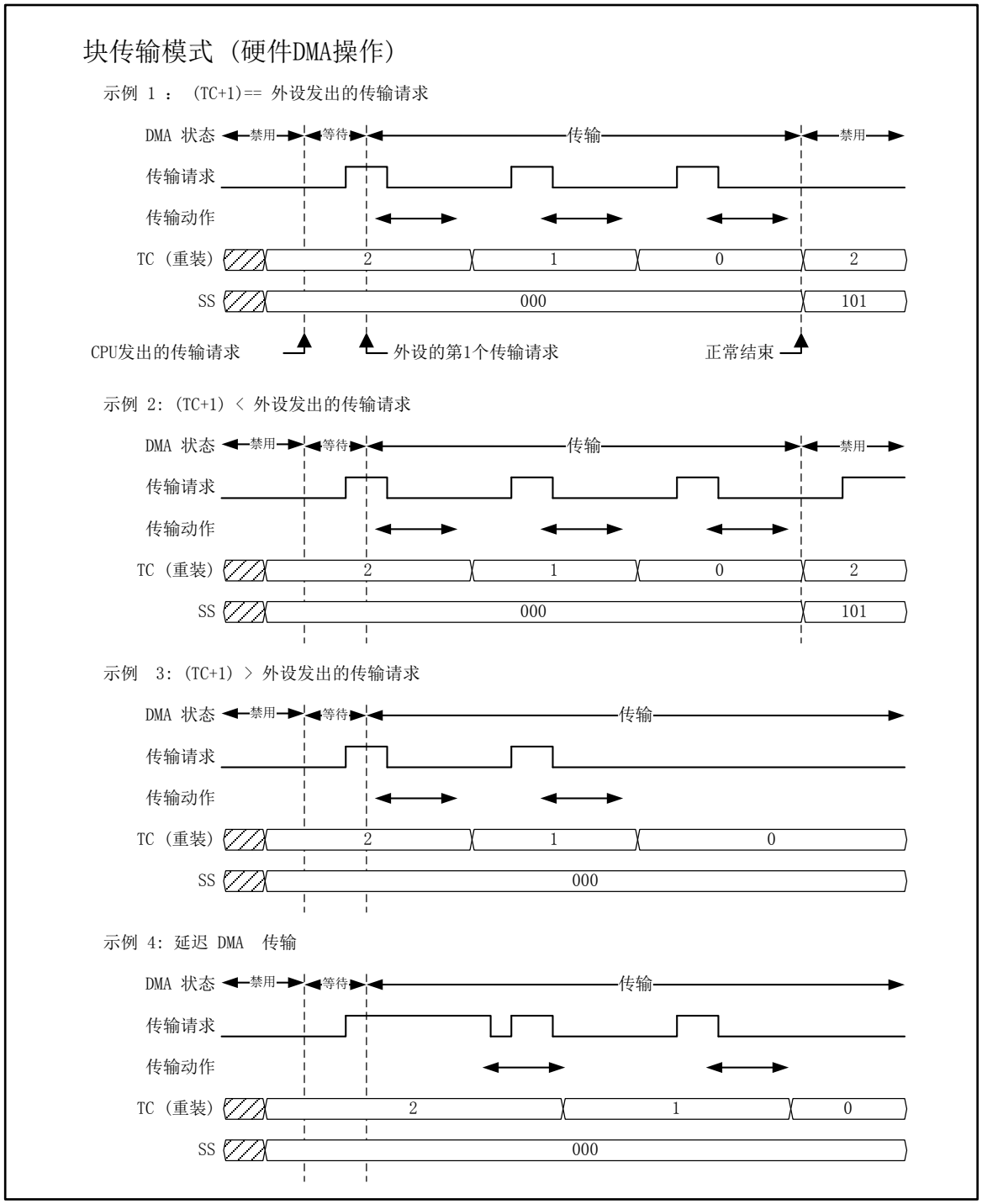


Figure 4-7 所示为块传输示例。块传输时，需要完成传输的传输请求数量为 TC+1。除非传输请求数量多于或少于请求数量，否则 CPU 不会介入（Figure 4-7 中的示例 1）。

Figure 4-7 硬件-块传输操作



如外设生成的传输请求数量超过 DMAC 的设置传输数量，指定传输数量完成后 DMAC 变为禁用状态。禁用状态下不会执行其他传输。同样，由于未从 DMAC 中清除多余的传输请求信号，将保持生效状态。这种情况下，解除来自 CPU 的传输请求信号（Figure 4-7 中的示例 2）。
如外设生成的传输请求数量少于 DMAC 的设置传输数量，DMAC 将保持传输状态并等待剩余数量的传输请求（Figure 4-7 中的示例 3）。
DMAC 的传输进程可能会太慢，不能跟上外设传输请求的生成间隔。块传输时，如 DMAC 传输进程滞后于外设的传输请求，传输操作时下一个传输请求信号的上升沿将被忽略。同样，在传输操作中，从 DMAC 中清除失效的传输请求信号。然后 DMAC 将保持传输状态并等待剩余的传输请求（Figure 4-7 中的示例 4）。
突发传输时，收到首次传输请求后，可访问系统总线时执行所有 (BC+1)×(TC+1) 的传输。所要求的外设传输请求数量仅为第一次请求。与块传输相同，生成的传输请求信号数量超过要求量时，则在禁用状态下被忽略。

5. 传输状态 =>禁用状态/传输成功完成

   参见软件传输控制的步骤 4。

6. 传输状态 =>禁用状态/传输错误停止

   参见软件传输控制的步骤 5。

7. 传输状态 =>禁用状态/外设停止请求结束

   如果外设传输停止信号生效，传输状态下的通道将暂停其传输进程。清除 EB、PB 及 ST 并转为禁用状态。将 SS[2:0] 设置为 "010" 并通知错误停止。通过 EI 使能中断时，发生传输失败中断。未指定重装的 BC、TC、DMACSA 及 DMACDA 保持传输暂停时设置的值。注意 SS[2:0] 值与软件停止请求的值相同。

8. 传输状态、暂停状态 =>禁用状态/强行终止传输

   参见软件传输控制的步骤 6。

9. 禁用状态/传输后的进程

   参见软件传输控制的步骤 7。
   外设停止请求、软件强行终止请求以及传输错误停止时，由于处理的传输数量少于外设的传输请求数量，通常保持传输请求信号有效。CPU 指示外设使传输请求信号无效。外设发出停止请求时，只要停止请求信号生效，传输请求信号将被屏蔽。也将使传输停止请求信号失效。
   即便 DMAC 成功完成指定的传输数量，传输请求信号会保持生效或失效，具体取决于外设的设置。注意这可能会影响下一次传输。

10. 传输状态、暂停状态/传输暂停

    参见软件传输控制的步骤 8。

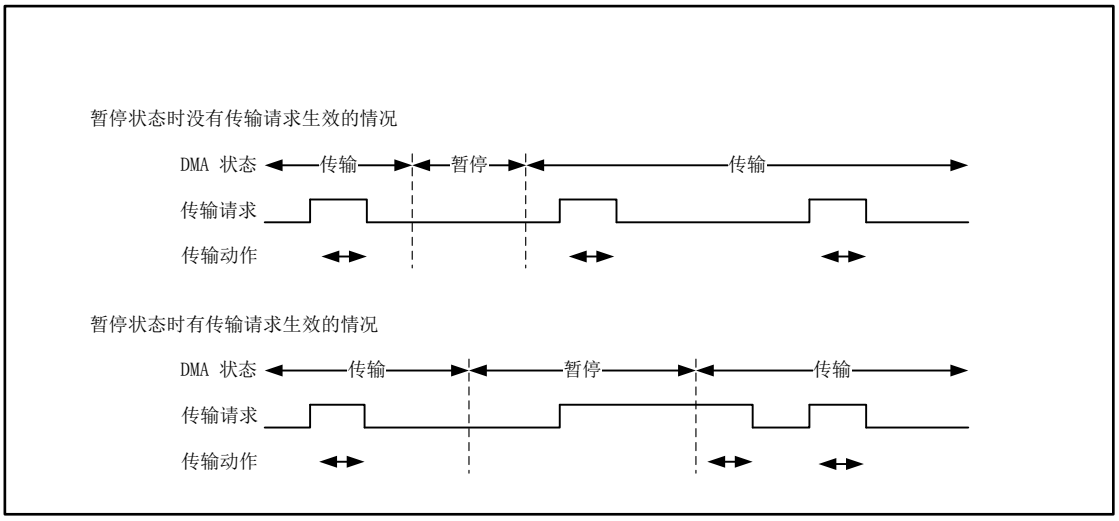
11. 暂停状态

    参见软件传输控制的步骤 9。
    即便外设的传输请求信号生效，暂停状态的通道也不会执行传输。也不会清除传输请求信号。

12. 暂停状态/传输暂停的取消

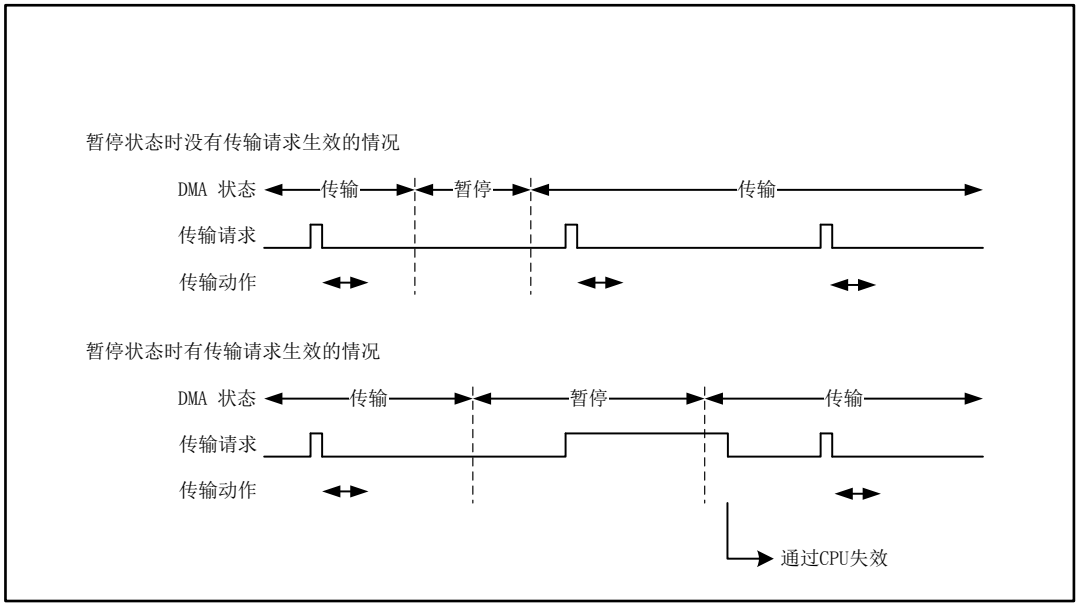
    参见软件传输控制的步骤 10。
    暂停状态下发出取消暂停指令时，返回传输状态。在先前的暂停状态下使传输请求信号生效时，如下所述，后续操作取决于传输模式。
    命令传输模式时，暂停状态下保持传输请求信号生效。因此，DMAC 返回传输状态时传输恢复，且正常清除传输请求信号。参见 Figure 4-8。

Figure 4-8 暂停状态下的命令传输操作



块传输模式时保持传输请求信号生效。即使返回传输状态，也不能检测出传输请求信号的上升沿，且不能恢复传输。所以暂停状态下传输请求被忽略。也不从 DMAC 清除传输请求信号。要恢复暂停的传输，向 DMAC 发出取消暂停的指令后，CPU 指令外设使传输请求信号失效。其后，外设生成下一个传输请求时传输恢复。此时，要注意外设输出的传输请求数量与 DMAC 收到的传输请求数量之间的差异。参见 Figure 4-9。

Figure 4-9 暂停状态下的块传输操作



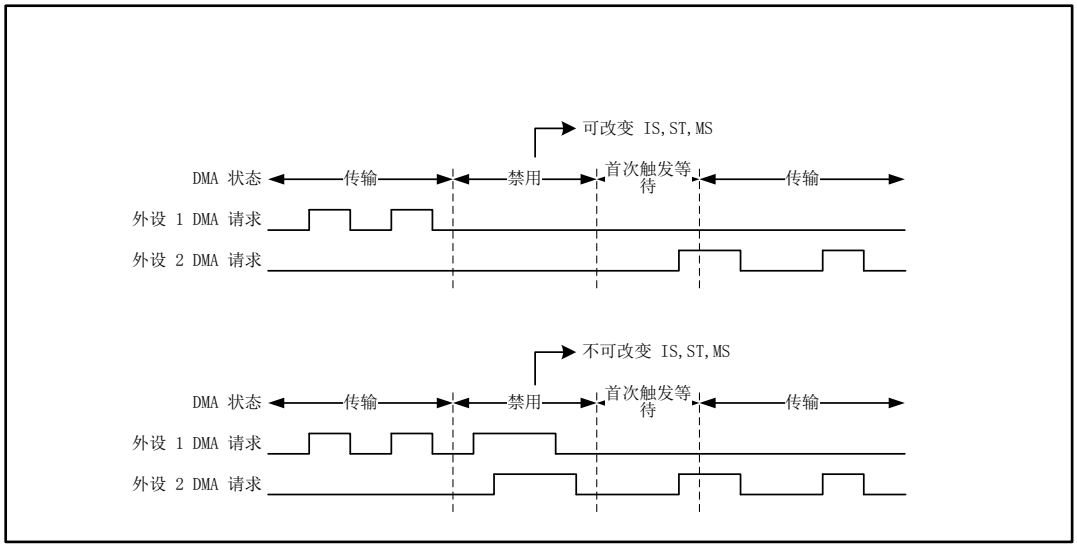
13. 禁用状态及首次触发等待状态的操作

参见软件传输控制的步骤 11。

如传输请求信号没有对禁用状态下的通道生效，可自由改变传输内容的指定（重写寄存器 DMACSA、DMACDA、DMACA[29:0] 及 DMACB）。

如果传输请求信号生效或者可能在禁用状态下生效，则不能改变 IS, ST 及 MS 中指定的传输内容。如尝试改变这些设置，DMAC 可能出现意外动作。改变 IS、ST 及 MS 的设置时，首先清除 CPU 向外设（改变之前或之后使用）发出的传输请求信号，然后在传输请求信号失效时改变设置。参见 Figure 4-10。

Figure 4-10 改变 IS、ST 及 MS 设置



不能从 CPU 改变首次触发等待状态通道的指定传输内容。

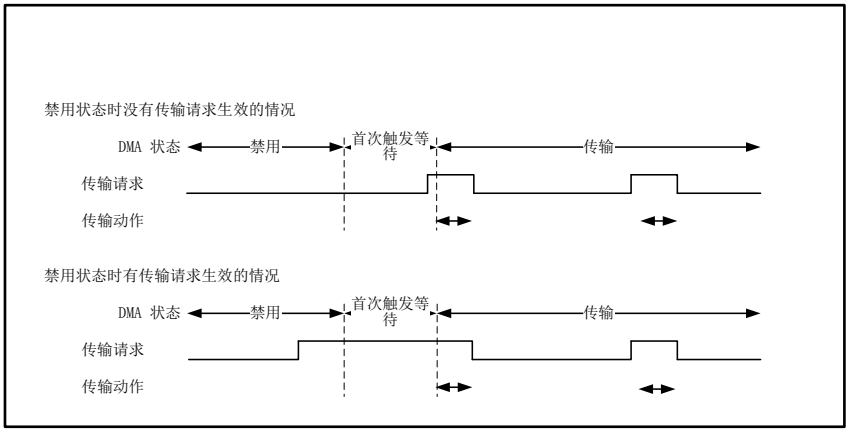
如传输请求信号对首次触发等待状态下的通道没有生效，CPU 发出禁用单通道或全部通道操作指令或暂停单通道或全部通道操作指令时，通道转为禁用状态。这种情况被视为已取消使能的传输。任何情况下 SS 都不会改变。

如果传输请求信号可以被等待首次触发状态的通道生效，那么应该注意的是，在 DMAC 已经开始传输或者完成传输之前，CPU 试图取消传输操作的使能。

禁用状态下，即便传输请求信号生效，DMAC 也不会启动传输或清除传输请求。当传输请求信号生效时，如果按照 CPU 的指令转换为等待首次触发状态，下列操作会适用（如前所述，仅在不改变 IS、ST 及 MS 的设置时适用）。

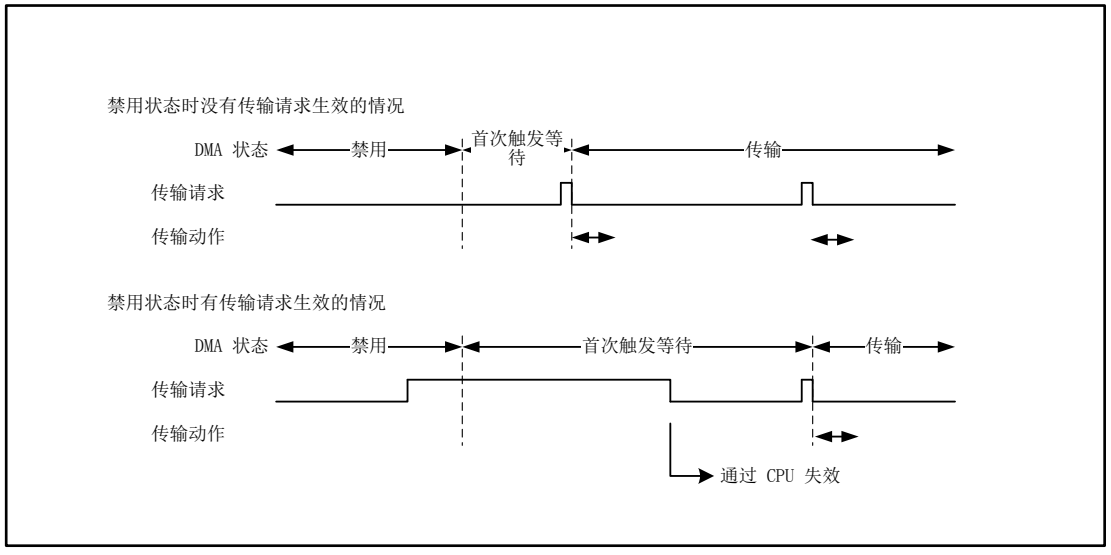
命令传输模式时，由于保持传输请求信号生效，DMAC 将立即转为传输状态并启动传输。正常地从 DMAC 中清除传输请求信号。参见 Figure 4-11。

Figure 4-11 禁用状态下的命令传输操作



块传输模式时保持传输请求信号有效。即使转为首次触发等待状态，也不能检测出传输请求信号的上升沿，不能恢复传输。所以禁用状态下传输请求被忽略。也不能从 DMAC 中清除传输请求信号。如果要恢复传输，需要指令 DMAC 转变为首次触发等待状态，然后根据 CPU 指令解除外设传输请求信号的生效。其后，外设生成下一次传输请求时，通道变为传输状态并恢复传输。此时，要注意外设输出的传输请求数量与 DMAC 收到的传输请求数量之间的差异。参见 Figure 4-12。

Figure 4-12 禁用状态下的块传输操作



- 补充说明 1

  参见软件传输控制的补充说明 1。
  硬件传输时， ST 始终写入 0 。

- 补充说明 2

  参见软件传输控制的补充说明 2。

- 补充说明 3

  参见软件传输控制的补充说明 3。

- 补充说明 4

  参见软件传输控制的补充说明 4。

- 补充说明 5

  如需要使外设传输请求信号（中断信号）失效，可采用以下方法：通常外设的中断信号是采用中断使能标志屏蔽（逻辑“与”）的中断因素标志。可通过复位任一标志使中断信号失效。复位后设置中断使能标志后时，中断信号出现上升沿。按此控制步骤可再次通知 DMAC 发出块传输的传输请求。有关详细信息，参见各外设《手册》。

4.4 硬件 (EM=1) 传输的 DMAC 操作及控制步骤

本节说明硬件 (EM=1) 传输的 DMAC 操作及控制步骤。

Figure 4-13 硬件 (EM=1) 传输状态转换图

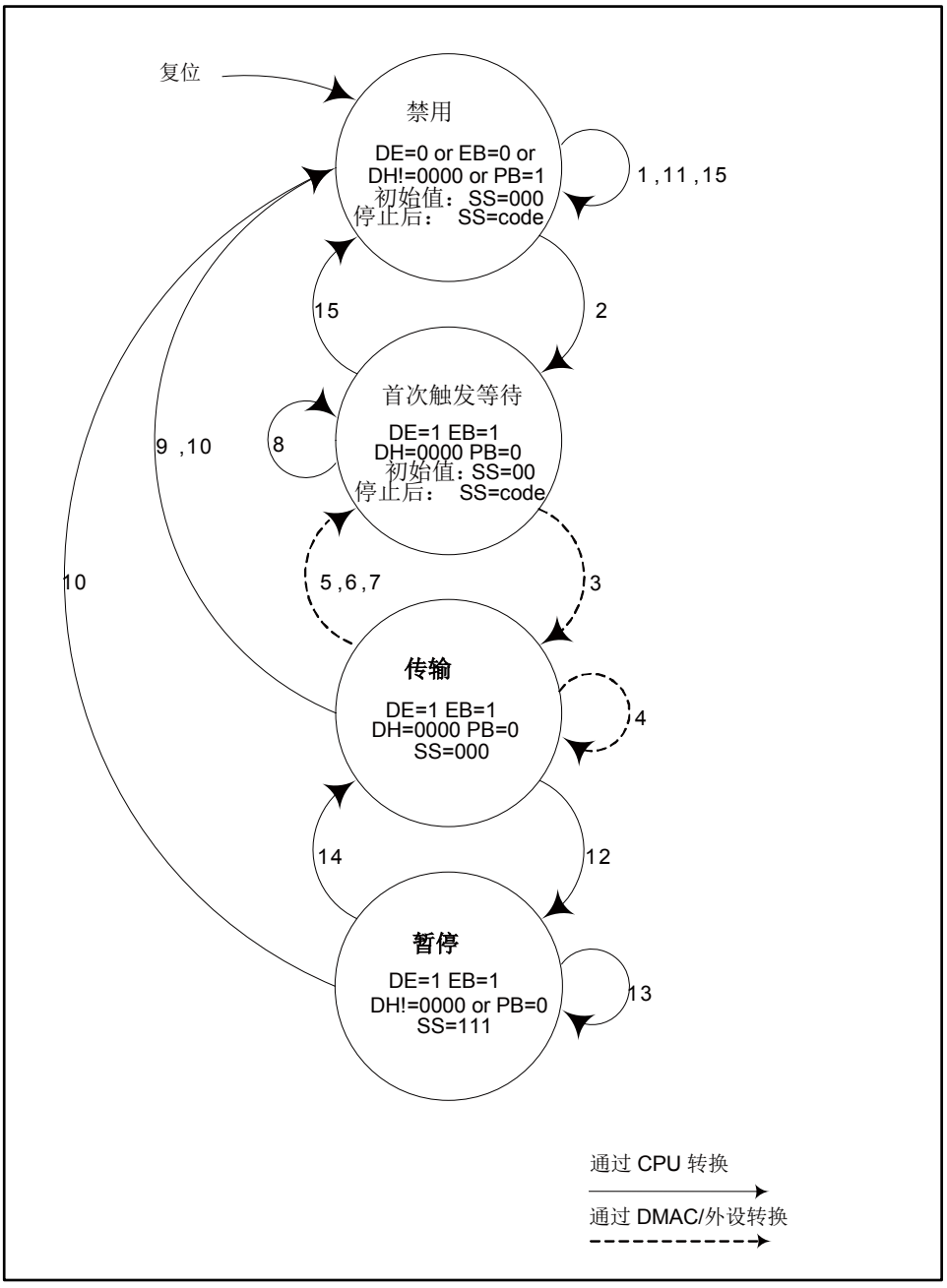


Figure 4-13 所示为硬件 (EM=1) 传输中受控通道的状态转换图。Figure 4-13 中转换线旁边的数字与下述控制步骤中出现的数字相对应。转换实线表示 CPU 指令的状态转换，而虚线表示 DMAC/外设操作引起的状态转换。

EM（使能位清除屏蔽）用于受控通道完成传输后屏蔽 EB 清除。EM=1 使能要重复的相同传输进程，无需 CPU 指令。

各状态描述

- 禁用状态

  参见硬件传输 (EM=0) 的控制。

- 首次触发等待状态

  参见硬件传输 (EM=0) 的控制。

- 传输状态

  此状态下，受控通道已收到外设的首次传输传输请求。此状态的通道执行指定的传输操作。当 EM=1，所有传输操作完成后，通道转为首次触发等待状态。CPU 发出指令后通道也会改变状态。

- 暂停状态

  参见硬件传输 (EM=0) 控制步骤。

控制步骤说明

1. 禁用状态/传输准备

   参见硬件传输 (EM=0) 控制的步骤 1。
   如果 EM=1，设置指定所有重装的传输内容（RC、RS、RD），防止意外地址域的数据传输。而且不设置 CI，因为对 DMAC 产生传输成功中断没有意义。设置 EI，从 DMAC 产生传输失败中断。

2. 禁用状态 =>首次触发等待状态/ 使能传输

   参见硬件传输 (EM=0) 控制的步骤 2。

3. 首次触发等待状态/传输启动

   参见硬件传输 (EM=0) 控制的步骤 3。

4. 传输状态

   参见硬件传输 (EM=0) 控制的步骤 4。

5. 传输状态 =>首次触发等待状态/传输成功

   (BC+1)× (TC+1) 计算的所有传输次数都成功完成后，传输状态的通道不清除 EB，而是清除 PB 和 ST，并转为首次触发等待状态。通道设置 SS[2:0]=101，通知传输成功完成。由于不设置 CI，所以不产生传输成功中断。因设置了 RC、RS 及 RD，重装 BC、TC、DMACSA 及 DMACDA 的指定传输内容。

6. 传输状态 =>首次触发等待状态/传输错误停止

   参见硬件传输 (EM=0) 控制的步骤 6。
   如果 EM=1，即便因错误停止传输，EB 也不会被清除。清除 PB 和 ST，转为首次触发等待状态，等待下一次传输请求。所以，可能发生传输错误的地址域里不建议通过 EM=1 使用 DMA 传输。

7. 传输状态 =>首次触发等待状态/外设停止请求结束

   参见硬件传输 (EM=0) 控制的步骤 7。
   如果 EM=1，即便外设发出停止请求，EB 也不会被清除。清除 PB 和 ST 并转为首次触发等待状态。因设置了 RC、RS 及 RD，重装 BC、TC、DMACSA 及 DMACDA 的指定传输内容。由于设置了 EI，产生传输失败中断。

8. 首次触发等待状态/传输后的进程

   如果 EM=1，传输完成后不会清除 EB。设置（DE=1、EB=1、DH=0000、PB=0）并转为首次触发等待状态。外设生成下一次传输请求时，启动下一次传输，无需 CPU 指令。
   如收到外设的停止请求后变为首次触发等待状态，则发生传输失败中断，并可确定该状态。停止请求信号生效时也会屏蔽传输请求信号。即便外设的下一次传输请求信号生效，该信号不会被识别，受控通道将保持首次触发等待状态，等待 CPU 发出指令。
   上述情况下，从 CPU 中读取 SS[2:0]，以检查传输的完成状态。从 CPU 中清除 SS[2:0]，使中断信号失效。CPU 清除 EB 并返回禁用状态（此为硬件传输 (EM=1) 控制步骤中步骤 15 所述操作）。如硬件传输 (EM=0) 控制步骤中步骤 7 所述，使外设的传输请求信号和停止请求信号失效。

9. 传输状态 =>禁用状态/ EM=0 时传输完成

   通过 CPU 写入 EM=0 可从首次触发等待状态和传输状态的循环中退出操作。收到指令停止传输计时期间，EB、ST 和 PB 被清除，传输状态变为禁用状态（DE=1、EB=0、DH=0000、PB=0），以成功完成传输。此时，因未设置 CI，不产生传输成功中断。

10. 传输状态、暂停状态 =>禁用状态/强行终止传输

    参见硬件传输 (EM=0) 控制的步骤 8。
    发出操作禁用指令可从首次触发等待状态和传输状态的循环中退出操作。发出禁用单通道操作指令后，相关通道转为禁用状态（DE=1、EB=0、DH=0000、PB=0）并停止操作。发出使能全部通道操作指令后，相关通道转为禁用状态（DE=0、EB=1、DH=0000、PB=0）并停止操作。发出禁用全部通道操作指令时，必须注意的是不清除 EB。
    若从传输状态中退出操作，因强行停止，传输失败，于是发生传输失败中断。从首次触发等待状态中退出时，使能的传输被取消（此操作如硬件传输控制步骤中步骤 15 所述）。

11. 禁用状态/传输后的进程

    参见硬件传输 (EM=0) 控制的步骤 9。

12. 传输状态、暂停状态/传输暂停

    参见硬件传输 (EM=0) 控制的步骤 10。

13. 暂停状态

    参见硬件传输 (EM=0) 控制的步骤 11。

14. 暂停状态/传输暂停的取消

    参见硬件传输 (EM=0) 控制的步骤 12。

15. 禁用状态及首次触发等待状态下的操作

    参见硬件传输 (EM=0) 控制的步骤 13。
    如果 EM=1，传输状态直接转为首次触发等待状态。因此，重复传输操作期间不能重写指定的传输内容（重写寄存器 DMACSA、DMACDA、DMACB[31:1] 和 DMACA[28:0]）。

- 补充说明 1

  参见硬件传输 (EM=0) 控制步骤中的补充说明 1。

- 补充说明 2

  参见硬件传输 (EM=0) 控制步骤中的补充说明 2。
  如果 EM=1，因传输操作时不清除 EB，补充说明 2 不适用。

- 补充说明 3

  参见硬件传输 (EM=0) 控制步骤中的补充说明 3。

- 补充说明 4

  参见硬件传输 (EM=0) 控制步骤中的补充说明 4。
  下文说明通过 EM=1 设置 DMAC 的中断时必须注意的事项。由于外设的停止请求产生的传输失败中断，不能改变通道的首次触发等待状态，所以中断信号不会被消除直到 CPU 的消除指令到来。同样，因软件的停止请求发生传输失败中断，目标通道转为禁用状态，CPU 清除中断信号前不会使中断信号失效。如相关通道转为传输状态，会在 CPU 未指示的时序使其他传输成功中断及传输失败中断失效。所以，对此必须引起注意。

- 补充说明 5

  参见硬件传输 (EM=0) 控制步骤中的补充说明 5。

5. DMAC 寄存器

本节说明各 DMAC 寄存器的功能。

5.1 寄存器列表
5.2 完整 DMAC 配置寄存器 (DMACR)
5.3 配置 A 寄存器 (DMACA)
5.4 配置 B 寄存器 (DMACB)
5.5 传输源地址寄存器 (DMACSA)
5.6 传输目标地址寄存器 (DMACDA)

5.1 寄存器列表

Table 5-1 列出 DMAC 控制寄存器。

Table 5-1 DMAC 控制寄存器列表

缩写	受控通道	寄存器名称	参考章节
DMACR	所有	完整 DMAC 配置寄存器	5.2
DMACA0	ch.0	配置 A 寄存器	5.3
DMACB0		配置 B 寄存器	5.4
DMACSA0		传输源地址寄存器	5.5
DMACDA0		传输目标地址寄存器	5.6
DMACA1	ch.1	配置 A 寄存器	5.3
DMACB1		配置 B 寄存器	5.4
DMACSA1		传输源地址寄存器	5.5
DMACDA1		传输目标地址寄存器	5.6
DMACA2	ch.2	配置 A 寄存器	5.3
DMACB2		配置 B 寄存器	5.4
DMACSA2		传输源地址寄存器	5.5
DMACDA2		传输目标地址寄存器	5.6
DMACA3	ch.3	配置 A 寄存器	5.3
DMACB3		配置 B 寄存器	5.4
DMACSA3		传输源地址寄存器	5.5
DMACDA3		传输目标地址寄存器	5.6
DMACA4	ch.4	配置 A 寄存器	5.3
DMACB4		配置 B 寄存器	5.4
DMACSA4		传输源地址寄存器	5.5
DMACDA4		传输目标地址寄存器	5.6
DMACA5	ch.5	配置 A 寄存器	5.3
DMACB5		配置 B 寄存器	5.4
DMACSA5		传输源地址寄存器	5.5
DMACDA5		传输目标地址寄存器	5.6
DMACA6	ch.6	配置 A 寄存器	5.3
DMACB6		配置 B 寄存器	5.4
DMACSA6		传输源地址寄存器	5.5
DMACDA6		传输目标地址寄存器	5.6
DMACA7	ch.7	配置 A 寄存器	5.3
DMACB7		配置 B 寄存器	5.4
DMACSA7		传输源地址寄存器	5.5
DMACDA7		传输目标地址寄存器	5.6

5.2 完整 DMAC 配置寄存器 (DMACR)

本节说明完整 DMAC 配置寄存器 (DMACR)

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	DE	DS	保留	PR	DH[3:0]				保留							
属性	R/W	R/W	-	R/W	R/W				-							
初始值	0	0	-	0	0000				-							

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留															
属性	-															
初始值	-															

[bit31] DE：DMA 使能（全部通道操作使能位）

此位控制全部通道传输操作的使能和禁用。

此位设置为 "1" 时，使能全部通道操作，各通道根据其设置执行操作。

此位设置为 "0" 时，禁用全部通道操作，将此位设置为 "1" 前不执行传输。正进行传输操作的通道也将强行停止传输。

使用此位强行停止当前正执行传输的全部通道并复位配置寄存器。

位	功能
0	禁用全部通道操作。（初始值）
1	使能全部通道操作。

[bit30] DS：DMA 停止

此位指示所有通道的传输状态。

如传输操作中设置了以下任一条件，DMAC 将此位设置为 "1"。

- 将 "0" 写入 DMACR:DE 位时，所有通道完成传输。
- 将 "0000" 之外的值写入 DMACR:DH 位时，所有通道暂停传输。

设置 DMACR:DE=1、DMACR:DH=0000 且所有通道使能操作时，DMAC 将此位设置为 "0"。

尽管此位的属性为 R/W，CPU 写入此位不会影响 DMAC 的操作。但若需要在不影响此位状态的前提下更新 DMACR 寄存器，首先读取此位，然后重新写入相同的值。

位	功能
0	清除全部通道操作禁用或全部通道暂停设置。（初始值）
1	因全部通道禁用或全部通道暂停设置，所有通道的传输停止。

[bit29] 保留：保留位

此位读取值为"0"。

写入此位时，将其设置为 "0"。

[bit28] PR：优先循环

此位控制通道的传输优先顺序。
此位设置为 "0" 时，所有通道的优先顺序固定。
此位设置为 "1" 时，使用循环方法确定所有通道的优先顺序。

位	功能
0	固定优先顺序。 (ch.0>ch.1>ch.2>ch.3>ch.4>ch.5>ch.6>ch.7)（初始值）
1	用循环方法确定优先顺序。

有关传输优先顺序的选择，参见 "3.5 通道优先控制"一节。

[bit27:24] DH：DMA 暂停（全部通道暂停位）

此位控制所有通道传输操作的暂停/取消。
此位设置为 "0000" 之外的值时，正执行传输的所有通道暂停传输。此位设置为 "0000" 时，传输恢复。

即使外部/外设设备的传输请求生效，暂停状态下的通道也会忽略该传输请求。使用块传输和突发传输时，即便清除了暂停，相关通道也不会启动传输。要完成传输时设置暂停的传输，需要在取消暂停后发出其他传输请求。

此位可用于暂停传输，无需复位所有通道的配置寄存器。

bit27:24	功能
0000	取消所有通道的传输暂停。（初始值）
0000 之外的值	暂停所有通道的传输。

[bit23:0] 保留：保留位

这些位的读取值为 "0"。
写入这些位时，将其设置为 "0"。

5.3 配置 A 寄存器 (DMACA)

本节说明配置 A 寄存器 (DMACA)。

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	EB	PB	ST	IS[5:0]						保留			BC[3:0]			
属性	R/W	R/W	R/W	R/W						-			R/W			
初始值	0	0	0	000000						-			0000			

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	TC[15:0]															
属性	R/W															
初始值	0x0000															

[bit31] EB：使能位（单通道操作使能位）

此位控制单通道传输操作的使能和禁用。

此位设置为 "1" 时，相关通道使能操作并等待触发，以使能其传输操作（必须将 DMACR:DE 设置为 "1"）。

如 EM 位 (DMACB[0]) 未设置为 "1"，传输完成后 DMAC 将此位清除为 "0"。

此位设置为 "0" 时，相关通道禁用操作，且在此位设置为 "1" 前不会执行传输操作。正进行传输操作的通道也将强行停止传输。

使用此位强行停止正执行传输的相关通道并复位配置寄存器。

位	功能
0	禁用相关通道操作。（初始值）
1	使能相关通道操作。

[bit30] PB：暂停位（单通道暂停位）

此位控制单通道传输操作的暂停/取消。

相关通道正执行传输操作时，将此位设置为 "1" 可暂停传输。此位设置为 "0" 时，传输恢复。

通道完成传输操作后，此位清除为 "0"。

即使外部/外设设备的传输请求生效，暂停状态下的通道也将忽略该传输请求。使用块传输和突发传输时，即便清除了暂停，相关通道也不会启动传输。要完成传输时设置暂停的传输，需要在取消暂停后发出其他传输请求。

使用此位可暂停传输，无需复位相关通道的配置寄存器。

位	功能
0	取消相关通道的传输暂停。
1	暂停相关通道的传输。

注意事项：

- *当 DMACB.RC="1" 时设置此位，必须同时设置 DMACA.BC 和 DMACA.TC，重装此位的字访问值。*

[bit29] ST：软件触发信号

此位用于生成单通道的软件传输请求。

此位设置为 "1" 时，通过软件传输请求触发，相关通道启动传输。传输完成后，DMAC 将此位清除为 "0"。若传输时将此位设置为 "0"，传输停止。

位	功能
0	无软件传输请求（初始值）
1	有软件传输请求

[bit28:23] IS[5:0]：输入选择

这些位选择传输请求的触发。

设置软件请求为传输触发 (ST=1) 时，将 IS[5:0] 位设置为 "000000"。
设置软件请求为传输触发时，指定用于启动传输的外设中断信号。可为所有通道选择任一外设。

连接 DMAC 的硬件传输请求信号根据所用产品而变化。选择设置前应检查连接至 "2.2DMAC 的 I/O 信号" 的传输请求。

bit28:23	功能
000000	软件(初始值)
100000	IDREQ[0]
100001	IDREQ[1]
100010	IDREQ[2]
100011	IDREQ[3]
100100	IDREQ[4]
100101	IDREQ[5]
100110	IDREQ[6]
100111	IDREQ[7]
101000	IDREQ[8]
101001	IDREQ[9]
101010	IDREQ[10]
101011	IDREQ[11]
101100	IDREQ[12]
101101	IDREQ[13]
101110	IDREQ[14]
101111	IDREQ[15]
110000	IDREQ[16]
110001	IDREQ[17]
110010	IDREQ[18]
110011	IDREQ[19]
110100	IDREQ[20]
110101	IDREQ[21]
110110	IDREQ[22]
110111	IDREQ[23]
111000	IDREQ[24]
111001	IDREQ[25]
111010	IDREQ[26]

bit28:23	功能
111011	IDREQ[27]
111100	IDREQ[28]
111101	IDREQ[29]
111110	IDREQ[30]
111111	IDREQ[31]
其他设置	禁止设置。

[bit22:20] 保留：保留位

这些位的读取值为 "0"。
写入这些位时，将其设置为 "0"。

[bit19:16] BC [3:0] ：块数

这些位指定块/突发传输的块数。

传输模式设置为命令传输时，将 BC[3:0] 设置为 "0000"。
设置“BC[3:0]= 块数 - 1"。最大允许块数为 16。
传输时可读取这些位的值。通常成功完成了一次传输源访问或传输目标访问后，BC[3:0] 减 1。

DMACB:RC=1 时，传输完成后重装启动传输时设置的值。
DMACB:RC=0 时，传输完成后此值设为 "0"，同时此值与传输失败暂停期间设置的值相同。

bit19:16	功能
	传输块数（初始值：0x0）

[bit15:0] TC[15:0]：传输数量

这些位指定块/突发/命令传输的传输数量。

设置“TC = 传输数量 – 1”。最大允许传输数量为 65536。

传输时可读取这些位的值。通常一个块传输完成后，TC 减 1。

DMACB:RC=1 时，传输完成后将重装启动传输时设置的值。
DMACB:RC=0 时，传输完成后将值设置为 "0"，该值与传输失败暂停期间设置的值相同。

bit15:0	功能
	传输数量（初始值：0x0000）

5.4 配置 B 寄存器 (DMACB)

本节说明配置 B 寄存器 (DMACB)。

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留		MS[1:0]		TW[1:0]		FS	FD	RC	RS	RD	EI	CI	SS[2:0]		
属性	R/W		R/W		R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
初始值	00		00		00		0	0	0	0	0	0	0	000		

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留															EM
属性	R/W															R/W
初始值	000000000000000															0

[bit31:30] 保留：保留位

这些位的读取值为 "0"。

写入这些位时，将其设置为 "0"。

[bit29:28] MS[1:0] ：模式选择

这些位选择传输模式。

位	功能
00	块传输模式（初始值）
01	突发传输模式
10	命令传输模式
11	保留

[bit27:26] TW[1:0]：传输位宽

这些位指定传输数据的位宽。

位	功能
00	字节（8 位）（初始值）
01	半字（16 位）
10	字（32 位）
11	保留

[bit25] FS：固定源

此位指定是否递增或固定传输源地址。

位	功能
0	根据 TW[1:0] 递增传输源地址。（初始值）
1	固定传输源地址。

[bit24] FD：固定目标

此位指定是否递增或固定传输目标地址。

位	功能
0	根据 TW[1:0] 递增传输目标地址。（初始值）
1	固定传输目标地址。

[bit23] RC：重装数（BC/TC 重装）

此位控制 BC[3:0] 及 TC[15:0] 的重装功能。

此位设置为 "1" 时，传输完成后将传输启动时设置的值重装至 BC[3:0] 和 TC[15:0]。

位	功能
0	禁用 BC/TC 的重装功能。（初始值）
1	使能 BC/TC 的重装功能。

[bit22] RS：重装源

此位控制传输源地址的重装功能。

此位设置为 "1" 时，传输完成后将传输启动时设置的值重装至 DMACSA。

位	功能
0	禁用传输源地址的重装功能。（初始值）
1	使能传输源地址的重装功能。

[bit21] RD：重装目标

此位控制传输目标地址 (DMACDA) 的重装功能。

此位设置为 "1" 时，传输完成后将传输启动时设置的值重装至 DMACDA。

位	功能
0	禁用传输目标地址的重装功能。（初始值）
1	使能传输目标地址的重装功能。

[bit20] EI：错误中断（传输失败中断使能）

此位使能或禁用传输失败时的中断通知。

此位设置为 "1" 时，若传输完成后 SS[2:0] 处于以下状态，则发出中断。

- 地址溢出
- 因外设的传输停止请求而停止，或 EB/DE 位禁用传输
- 传输源访问错误
- 传输目标访问错误

位	功能
0	禁用传输失败后发出的中断。（初始值）
1	使能传输失败后发出的中断。

[bit19] CI：完成中断：（传输成功中断使能）
此位使能或禁用传输成功时的中断。

此位设置为 "1" 时，如传输完成后将 SS[2:0] 设置为传输成功，则产生中断。

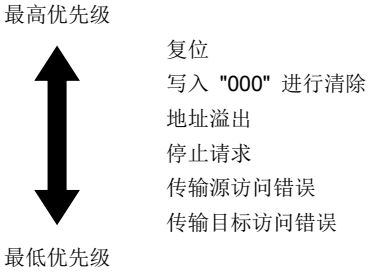
位	功能
0	禁用传输成功后发出的中断。（初始值）
1	使能传输成功后发出的中断。

[bit18:16] SS[2:0]：停止状态（停止状态通知）
这些位是指示传输停止状态或完成状态的代码。

下表列出可用代码。
如发出传输成功中断或传输失败中断，将 "000" 写入这些位可撤销该中断信号。

bit18:16	描述
000	初始值
001	传输错误终止（地址溢出）
010	传输停止请求终止（因外设发出的传输停止请求或 EB/DE 位禁用传输而终止）
011	传输错误终止（传输源访问错误）
100	传输错误终止（传输目标访问错误）
101	传输成功
110	保留
111	暂停传输

如同时发生多种错误，则根据以下优先顺序显示终止代码。



[bit15:1] 保留：保留位
这些位的读取值为 "0"。
写入这些位时，将其设置为 "0"。

[bit0] EM：使能位屏蔽（EB 位清除屏蔽）
此位用于传输完成后屏蔽通过 DMAC 对 EB 位 (DMACA[31]) 的清除。

如果 EM=0，传输完成后 DMAC 将 EB 位 (DMACA[31]) 清除为 "0"。
如果 EM=1，传输完成后 DMAC 不清除 EB 位。此功能允许不经 CPU 指令进行重复传输。

此功能只能用于硬件传输。使用此功能时，使能 RC、RS 及 RD 位的重装功能。

位	功能
0	传输完成后将 DMACA:EB 清除为 0。（初始值）
1	传输完成后不清除 DMACA:EB。

5.5 传输源地址寄存器 (DMACSA)

本节说明传输源地址寄存器 (DMACSA)。

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	DMACSA[31:16]															
属性	R/W															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	DMACSA[15:0]															
属性	R/W															
初始值	0x0000															

[bit31:0] DMACSA[31:0]：DMAC 源地址

这些位指定传输源的传输启动地址。

未对齐地址不能设置用于传输数据宽 (TW[1:0])。传输时可读取这些位的值。

DMACB:FS=1 时，将传输源地址设置为固定值，不变化。

DMACB:FS=0 且 DMACB:RS=0 时，根据 TW[1:0] 递增该值。

传输成功后，该地址为传输完成地址之后的下一个地址。

传输失败后，该值为暂停时设置的值。

DMACB:FS=0 且 DMACB:RS=1 时，传输时根据 TW[1:0] 递增该值。传输完成后，重装传输启动时设置的值。

	功能
bit31:0	指定传输启动的传输源地址。（初始值：0x00000000）

5.6 传输目标地址寄存器 (DMACDA)

本节说明传输目标地址寄存器 (DMACDA)。

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	DMACDA[31:16]															
属性	R/W															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	DMACDA[15:0]															
属性	R/W															
初始值	0x0000															

[bit31:0] DMACDA[31:0]：DMAC 目标地址

这些位指定传输目标的传输启动地址。

未对齐地址不能设置用于传输数据宽 (TW[1:0])。传输时可读取这些位的值。

DMACB:FD=1 时，将传输目标地址设置为固定值，不变化。

DMACB:FD=0 且 DMACB:RD=0 时，根据 TW[1:0] 递增该值。

传输成功后，该地址为传输完成地址之后的下一个地址。

传输失败后，该值为暂停时设置的值。

当 DMACB:FD=0 且 DMACB:RD=1，传输时根据 TW[1:0] 递增该值。

传输完成后，重装传输启动时设置的值。

bit31:0	功能
	DMA 传输启动的传输目标地址（初始值：0x00000000）

6. 使用注意事项

本节说明 DMAC 的使用注意事项

寄存器设置注意事项

设置 DMAC 寄存器时，请注意以下几点：

- 可字节、半字和字访问 DMACR、DMACA、DMACB、DMACSA 和 DMACDA 寄存器。
- 不能将 DMAC 中的寄存器地址设置至 DMACSA 或 DMACDA 寄存器。
- DMA 传输期间不能改变通道设置寄存器，但 DMACR 的 DE/DH 位、DMACA 的 EB/PB 位及 DMACB 的 EM 位除外。

停止模式和计时器模式转换时的注意事项

转换至停止模式和计时器模式时，务必停止所有 DMAC 通道操作，并确认使用 DS 标志停止 DMAC。若 DMAC 操作时转换至停止模式和计时器模式，返回至运行模式时可能出现意外操作。

第 10-1 章: I/O 端口

本章说明 **I/O** 端口。

1. 概述
2. 配置、框图及操作
3. 设置步骤示例
4. 寄存器
5. 使用注意事项

代码：9BFGPIO-FM0-C03.0

1. 概述

本节概述 I/O 端口。

本系列 I/O 端口具有以下功能特性：

- 本系列 I/O 端口复用以下功能：
  - GPIO

    通用 I/O 端口，可读取输入电平并设置 CPU 输出电平。
  - 快速 GPIO

    快速 GPIO，可在 1 个周期中读取输入电平并设置 CPU 输出电平。有关详细说明，参见“快速 GPIO”一章。
  - 外设输入/输出

    外设功能的数字输入/输出信号端口。
  - 专用 I/O 端口
  - 模拟输入端口

    A/D 转换器和 LCD 控制器模拟输入端口。
  - 模拟输出端口

    A/D 转换器和 LCD 控制器模拟输出端口。
  - USB 端口
  - 振荡端口
- 可对各引脚进行以下设置：
  - 可设置 I/O 端口用作 GPIO、外设功能数字引脚、或专用引脚。
  - 可设置 I/O 端口用作输入端口或输出端口。
  - 可使能或禁用上拉。
  - 将外设功能分配给两个或多个 I/O 端口，输入/输出此功能。可设置将功能（重定位功能）分配给某个 I/O 端口。
  - 可通过寄存器设置，CPU 待机模式时，将 I/O 端口设置为 Hi-Z 状态。

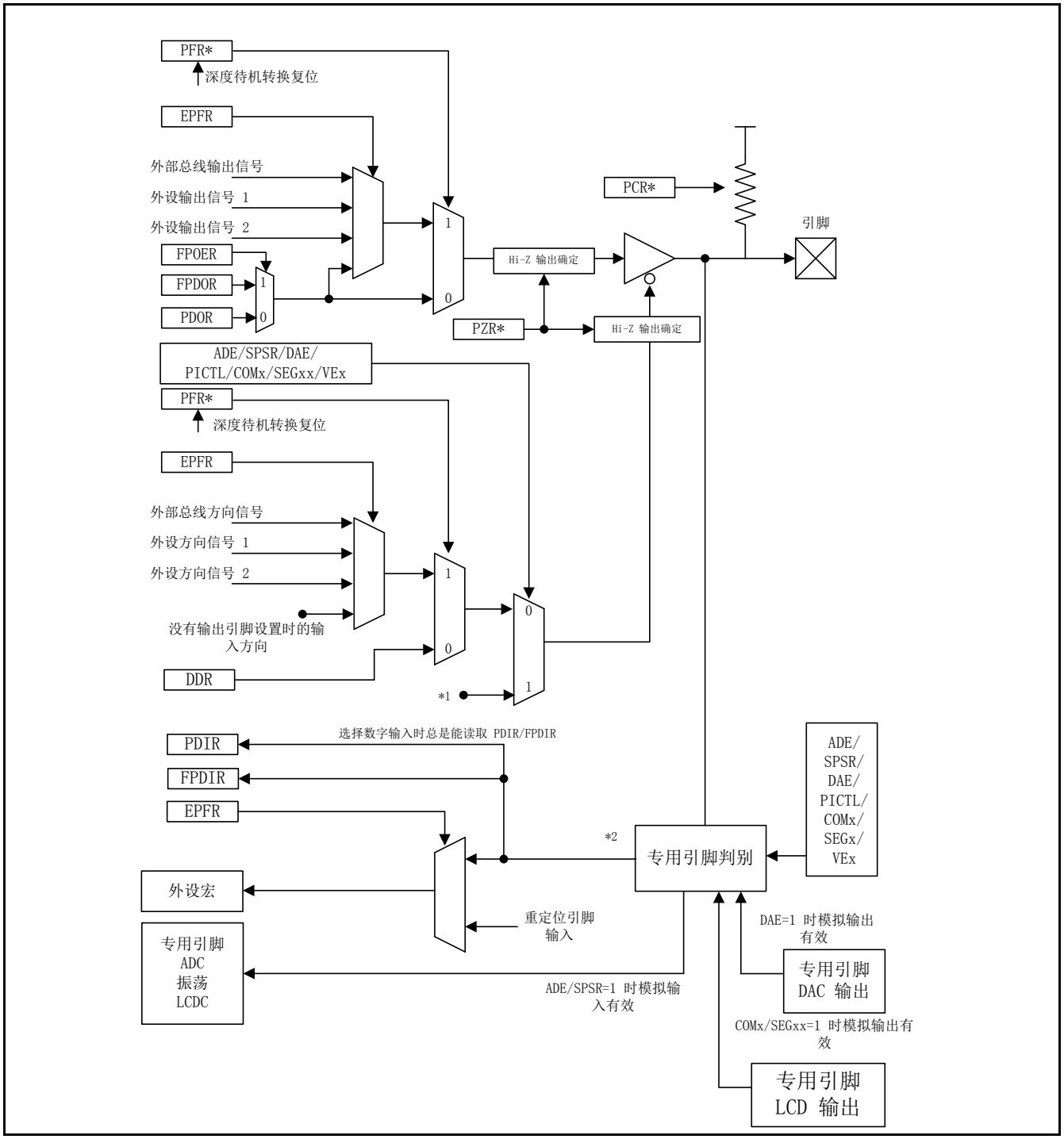
2. 配置、框图及操作

本节说明 I/O 端口配置、框图及操作。

2.1 I/O 端口配置

设置 I/O 端口寄存器，选择输入/输出方向并选择 GPIO/外设。
Figure 2-1 详细说明 I/O 端口。

Figure 2-1I/O 端口框图



*1: 以下任意设置将 I/O 端口设置为输入方向。

- ADE/SPSR=1
- DAE=1
- PICTL=0
- PICTL=1 且 COMx/SEGxx=1
- VEx=1

*2: 以下任意设置将输入值固定为 "0"。

否则，引脚将被设置为数字输入引脚。

- ADE/SPSR=1
- DAE=1
- PICTL=0
- PICTL=1 且 COMx/SEGxx=1
- VEx=1

注意事项:

- *某些产品中耐 5V 的 I/O 没有上拉电阻器。*
- *USB 引脚没有上拉电阻器。*
- *如果没有上拉电阻器，PCR 寄存器的设置无效。*
- *只有某些专用引脚能执行 PZR 寄存器功能。*
  只有《数据手册》"I/O 电路类型" 注释栏中 "使能 PZR 寄存器控制" 所述的引脚能控制本功能。
- *有关 DAE 位的详细说明，参见《模拟宏部分》中 "10 位 D/A 转换器" 内的 "5.1. D/A 控制寄存器 (DACR)"。*
- *有关 PICTL/COMx/SEGxx/VEx 位的详细说明，参见《模拟宏部分》中 "LCDC 控制器" 内的 "5.3 LCDC 控制寄存器 3 (LCDCC3)、"5.5 LCDC COM 输出使能寄存器 (LCDC_COMEN)" 和 "5.6 LCDC SEG 输出使能寄存器 1/2 (LCDC_SEGEN1/2)"。*
- *FPDIR/FPDOR/FPOER 为快速 GPIO 寄存器。有关详细说明，参见 "快速 GPIO" 一章。*

Table 2-1 列出寄存器功能。

- PFR、DDR、PDIR、PDOR 和 PCR 寄存器为各 I/O 端口配备 1 位控制寄存器，并为 I/O 端口选择功能。
- ADE 寄存器为各复用为模拟输入引脚的 I/O 端口配备 1 位控制寄存器，并为 I/O 端口选择功能。
- SPSR 寄存器选择用作 USB 引脚或振荡引脚的 I/O 端口选择功能。
- EPFR 寄存器为外设功能的各 I/O 引脚配备控制寄存器，并选择将外设功能 I/O 引脚重定位到哪一个 I/O 端口。
- PZR 寄存器设置通过 Hi-Zing I/O 端口控制的特定引脚高电平输出时的伪模式开漏控制。

Table 2-1 寄存器功能描述

寄存器名称	功能描述
ADE	将 I/O 端口设置为专用引脚（模拟输入引脚）或数字输入/输出引脚的寄存器。
SPSR	将 I/O 端口设置为专用引脚（USB 或振荡）或数字输入/输出引脚的寄存器。
PFR	将 I/O 端口设置为 GPIO 功能输入/输出引脚或外设功能输入/输出引脚的寄存器。
PCR	I/O 端口用作数字输入引脚或数字双向引脚时，用于设置连接或断开 I/O 端口上拉电阻器的寄存器。
DDR	I/O 端口用作 GPIO 功能引脚时，用于将 I/O 端口设置为输入引脚或输出引脚的寄存器。 注意事项：如果引脚被选为外设功能 I/O 引脚，则设置值无效。
PDIR	读取 I/O 端口电平状态的寄存器。 – 当 I/O 端口用作数字输入引脚时，本寄存器读取输入电平。 – 当 I/O 端口用作数字输出引脚时，本寄存器读取输出电平。 – 当 I/O 端口用作模拟输入引脚时，读取值始终为 "0"。
PDOR	当 I/O 端口用作 GPIO 功能输出引脚时，用来设置输出电平的寄存器。 – 设置为 "0" 时，输出低电平。 – 设置为 "1" 时，输出高电平。 注意事项：如果引脚被选为 GPIO 输入或外设功能输入/输出，则设置值无效。
EPFR	选择外设功能输入/输出功能并设置重定位功能的寄存器。 – 设置外设输出引脚 设置是否产生 I/O 端口的输出。此外，还可设置将外设功能各引脚重定位到某个 I/O 端口。 – 设置外设输入引脚 可设置将外设功能各引脚重定位到某个 I/O 端口。 – 设置外设双向引脚 可设置将外设功能各引脚重定位到某个 I/O 端口。
PZR	本寄存器设置 I/O 端口的开漏控制。 – 当 I/O 端口输出低电平时（无论 PCR 设置值如何，上拉断开），将 I/O 端口设置成低输出 – 当 I/O 端口输出高电平时（无论 PCR 设置值如何，上拉断开），通过将 I/O 端口设为 Hi-Z 状态，从而设置伪模式开漏控制 – 当 I/O 端口用于输入时（无论 PCR 设置值如何，上拉断开），将 I/O 端口设置成 Hi-Z 状态 注意事项： 只能某些专用引脚执行本功能。 只有《数据手册》“I/O 电路类型”注释栏中“使能 PZR 寄存器控制”所述的引脚能控制开漏。

Table 2-2 列出主要引脚功能，这些引脚的有效性由所选择的 I/O 端口功能以及寄存器设置值确定。

Table 2-2 I/O I/O 端口功能和寄存器设置值

I/O 端口功能		ADE/ SPSR/ DAE/ COMx/ SEGx/ VEx	PFR	DDR	PZR	PCR	EPFR
可用主要功能	可用子功能						
专用引脚 （模拟输入、 模拟输出、 USB、 振荡）	不适用	1	-	-	-	断开	*0
GPIO 功能输入引脚	外设功能输入引脚*5	0	0	0	0	有效	*1
				0	1	断开	
GPIO 功能输出引脚	GPIO 功能输入引脚 (FB) 外设功能输入引脚 (FB)			1	0	断开	
				1	1	断开	
外设功能输出引脚	GPIO 功能输入引脚 (FB) 外设功能输入引脚 (FB)		1	-	0	断开	*2
					1	断开	
外设功能双向引脚	GPIO 功能输入引脚 (FB) 外设功能输入引脚 (FB)				0	有效	*3
					1	断开	
外设功能输入引脚	GPIO 功能输入引脚				0	有效	*4
					1	断开	

图例

-: 表示寄存器设置值不影响引脚功能。

有效： 表示当 PCR 寄存器值为 0 时，上拉电阻器断开。
当 PCR 寄存器值为 1 时，上拉电阻器连接。

断开： 表示无论 PCR 寄存器值如何，上拉电阻器都断开。

(FB)： 表示反馈 I/O 端口的输出信号，并且通过 PDIR 读取 I/O 端口电平。此信号还可用作外设功能输入。

*0: 如果选择 I/O 端口为外设功能输入引脚，则设置无效。
如果选择 I/O 端口为外设功能输出引脚，则设置无效。
如果选择 I/O 端口为外设功能双向引脚，则设置无效。

*1: 如果选择 I/O 端口为外设功能输入引脚，则设置有效。
如果选择 I/O 端口为外设功能输出引脚，则设置无效。
如果选择 I/O 端口为外设功能双向引脚，则设置无效。

*2: 表示选择 I/O 端口为外设功能输出引脚。

*3: 表示选择 I/O 端口为外设功能双向引脚。

*4: 表示 I/O 端口既不用作不得被选为外设功能输出引脚也不用作双向引脚。

*5: 使用 NMIX 引脚时，设置 NMIS="1" 和 PFR="1"。

2.2 I/O 端口初选功能

Table 2-3 说明解除复位后各 I/O 端口的初选功能。

Table 2-3 复位解除后各 I/O 端口的初选功能

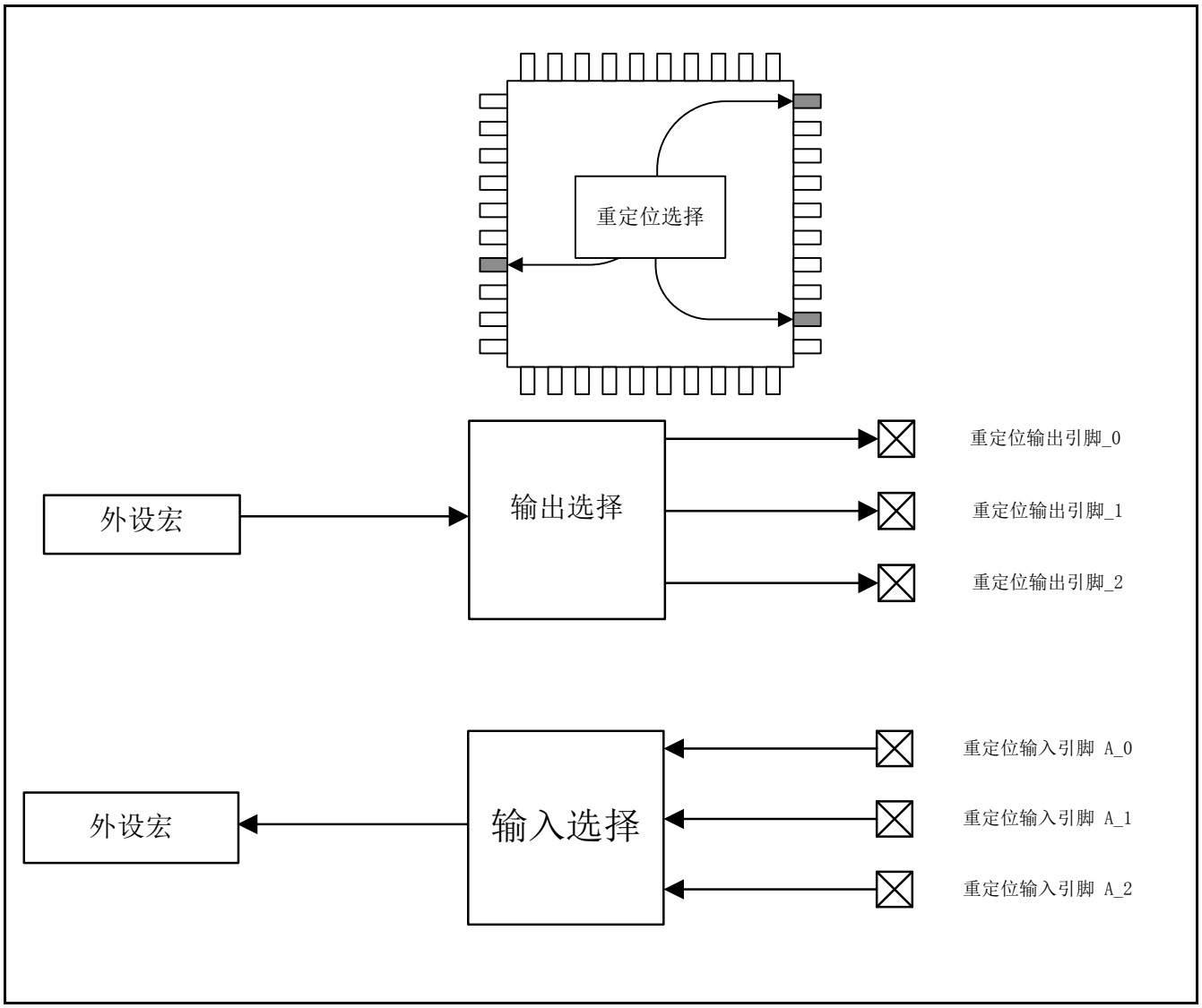
编号	引脚	初选功能
1	SWCLK, SWDIO	选择串行线调试 (SWD) 引脚。使能上拉
2	ANxx	可用作模拟输入引脚。数字输入被切断，输入 "0"。
3	X0,X1,X0A, X1A	可用作振荡引脚。数字输入被切断，输入 "0"。
4	上述引脚以外的所有 GPIO 引脚	数字输入。输出为 Hi-Z。

注意事项： 有关 GPIO（MD 引脚，一种复位引脚）以外的引脚状态，参见所使用产品的《数据手册》。在复位进程中，EPFR 的所有输出选择值为 "无输出"。

2.3 重定位功能

- 某些外设功能输入/输出具有的引脚（重定位引脚）不止一个。
  可设置 EPFR 选择其中一个引脚。Figure 2-2 所示为重定位功能原理图。

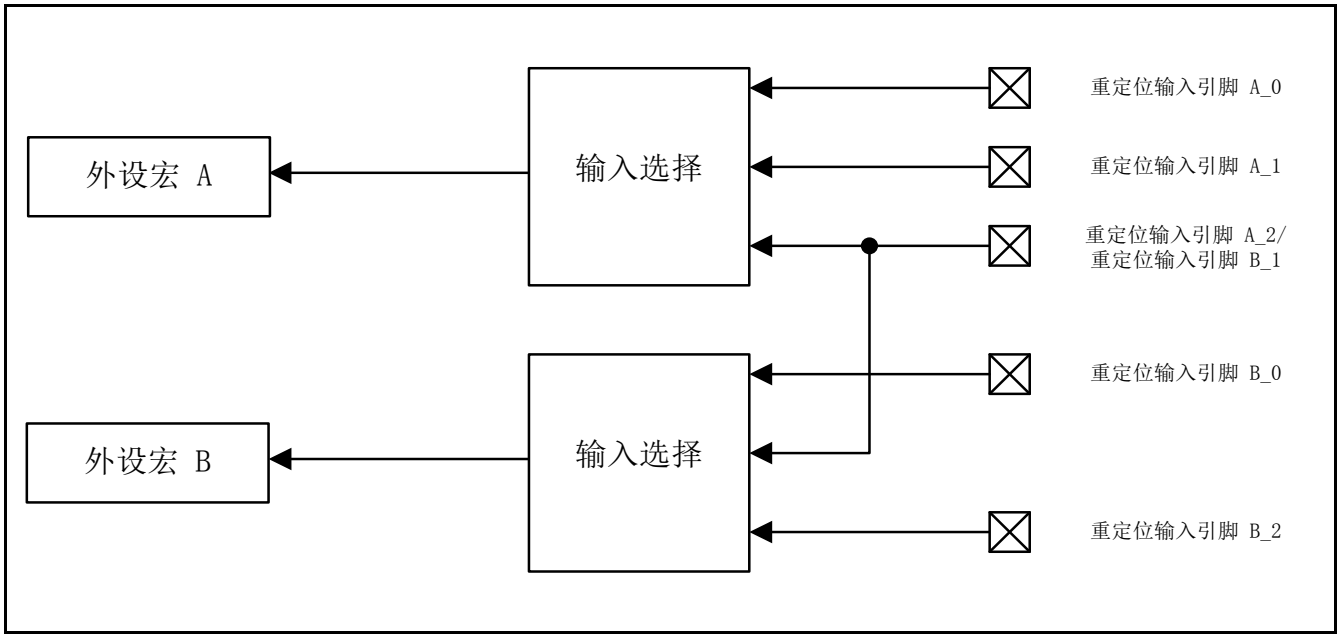
Figure 2-2 重定位功能原理图



注意事项： 哪种外设功能分配给哪一个引脚取决于产品。
参见所使用产品《数据手册》的引脚功能列表。

- 即使设置一个 I/O 端口输入相连两个或多个外设功能，也可通过设置 EPFR 使用所有外设输入。如 Figure 2-3 所示，可通过选择 "重定位输入引脚 A_2" 和 "重定位输入引脚 B_1" 输入实现同时使用所有外设输入。这样，可同时使用该 I/O 端口复用的外部中断和多功能串口引脚。

Figure 2-3 多路外设输入



- 即使 I/O 引脚设置为输出引脚，仍然可用作输入引脚，因为输入没有被屏蔽。例如，计时器输出可用作复用的外部中断输入。

2.4 EPFR 输出的固定优先级

在两个或多个输出中，只有一项输出引脚功能被分配给一个 I/O 端口。
设置 EPFR 寄存器时果设置了一个以上输出，则适用固定优先级并选择输出引脚。Figure 2-4 所示为输出引脚和固定优先级。

Figure 2-4 输出引脚和固定优先级

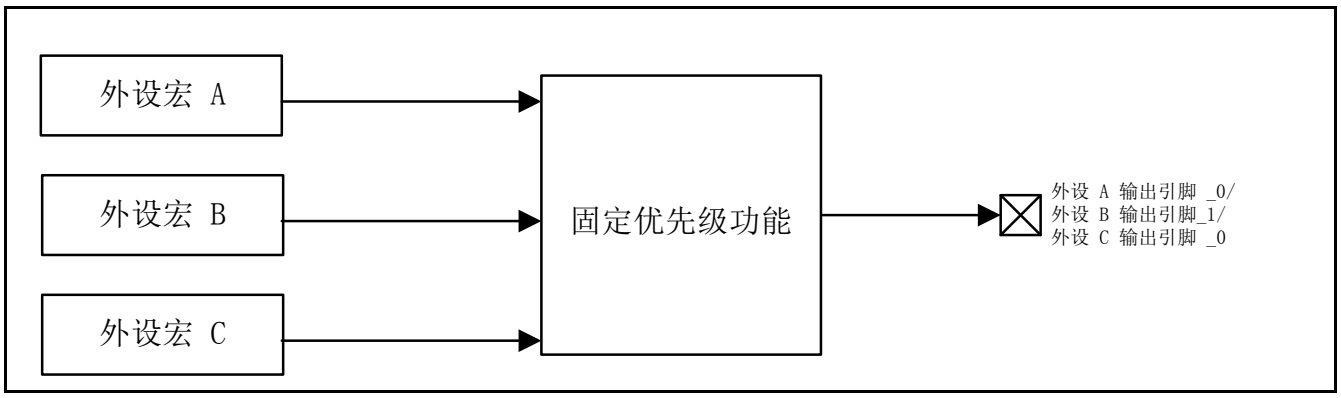


Table 2-4 说明 EPFR 的固定优先级。

Table 2-4EPFR 的固定优先级

优先序列	外设功能	适用引脚
高优先级	专用输入	串行线调试输入，NMI 输入
↓	串行线调试	I/O 引脚
↓	HDMI-CEC	I/O 引脚*2
↓	I2C SLAVE	I/O 引脚
↓	USB (HCONX)	输出引脚
↓	MFSI2S	输出引脚
↓	多功能串口	输出引脚，I/O 引脚*1
↓	基本计时器输出	I/O 引脚
↓	多功能计时器	输出引脚
↓	智能卡	输出引脚
↓	内部高速 CR 振荡时钟输出	输出引脚
↓	RTCCO 输出	输出引脚
低优先级	SUBOUT 输出	输出引脚

注意事项： 只有当一个以上的功能设置为 "输出" 时，固定优先级才适用。若为 "输入"，则无固定优先级。然而，"专用输入" 的优先级高于其他任何 "输出" 设置。如果设置了"专用输入" ，分配给同一端口的 "输出" 设置无效。

*1: 在 TYPE 1-M0+ 产品中，优先级为 SOT（串行数据输出）> SCS（芯片选择输出）。

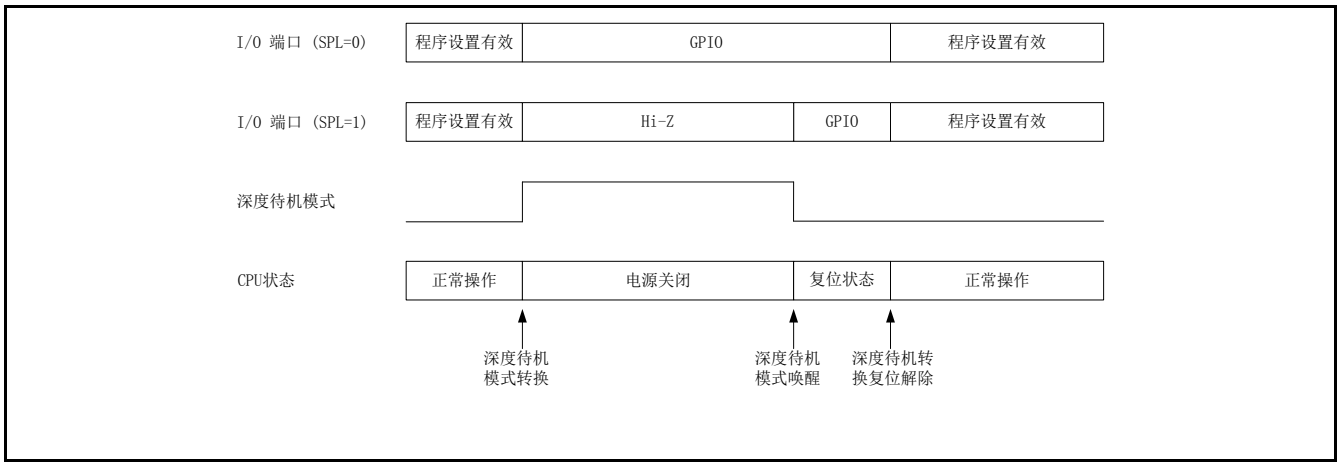
*2: 在 TYPE 3-M0+ 产品中，HDMI-CEC 引脚不受 EPFR 寄存器控制。

- 由于输出设置为较低优先级，EPFR 寄存器始终包括 "无输出" 设置。
- 如果要将引脚用作外设功能的外部输入引脚，则应禁用所有复用输出设置。如果 EPFR 寄存器没有选择所有引脚输出，则该引脚将用作外部输入引脚。

2.5 深度待机模式（TYPE 2-M0+）下的操作

在深度待机模式下选择 GPIO 功能。Figure 2-5 所示为 I/O 端口在深度待机模式下的操作。

Figure 2-5 深度待机模式时的 I/O 端口操作（TYPE 2-M0+产品）



注意事项：

- *部分端口没有选择 GPIO 功能。有关各引脚在深度待机模式下的状态，参见所使用产品《数据手册》中的引脚状态表。*

2.6 深度待机模式（TYPE 3-M0+产品）时的操作

在深度待机模式之初， I/O 端口选择为 GPIO 功能。然后，通过 CHOLDX 在 I/O 电路模块所含锁存电路中保持 I/O 端口的状态。

Figure 2-6 概述 I/O 电路模块。

Figure 2-6I/O 电路模块（TYPE 3-M0+产品）

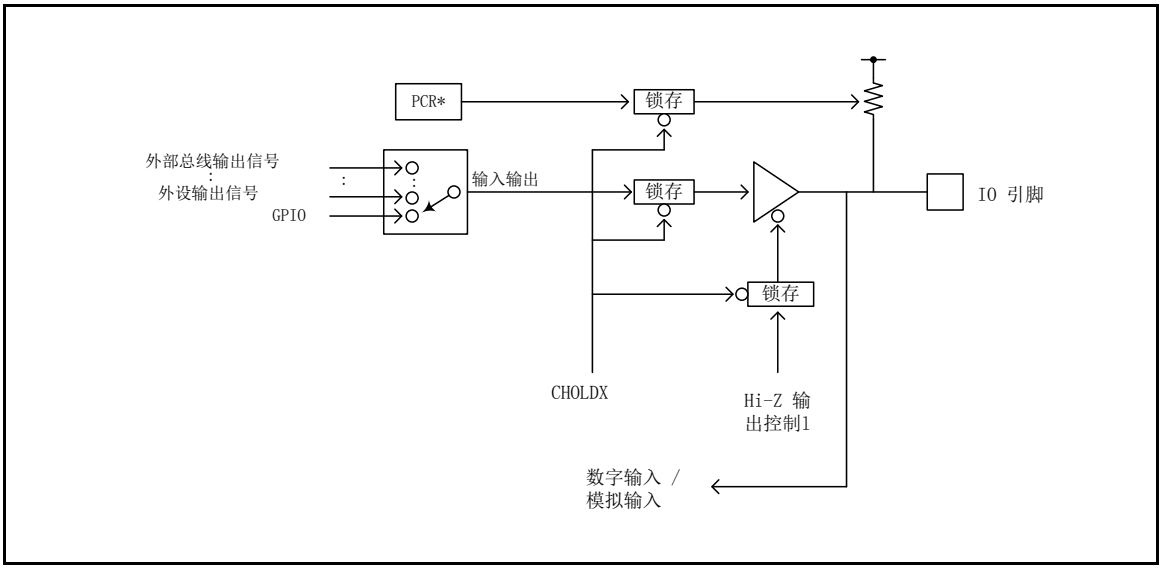


Figure 2-7 所示为 WIOLC_CTL:CONTX=1 时 I/O 端口在深度待机模式时的操作。在本示例中，I/O 端口状态在深度待机模式持续期内为 GPIO 功能状态。从深度待机模式唤醒后，将初始化 I/O 端口状态，如 Table 2-3 所示。

Figure 2-7 深度待机模式时的 I/O 端口操作（CONTX=1，TYPE 3-M0+产品）

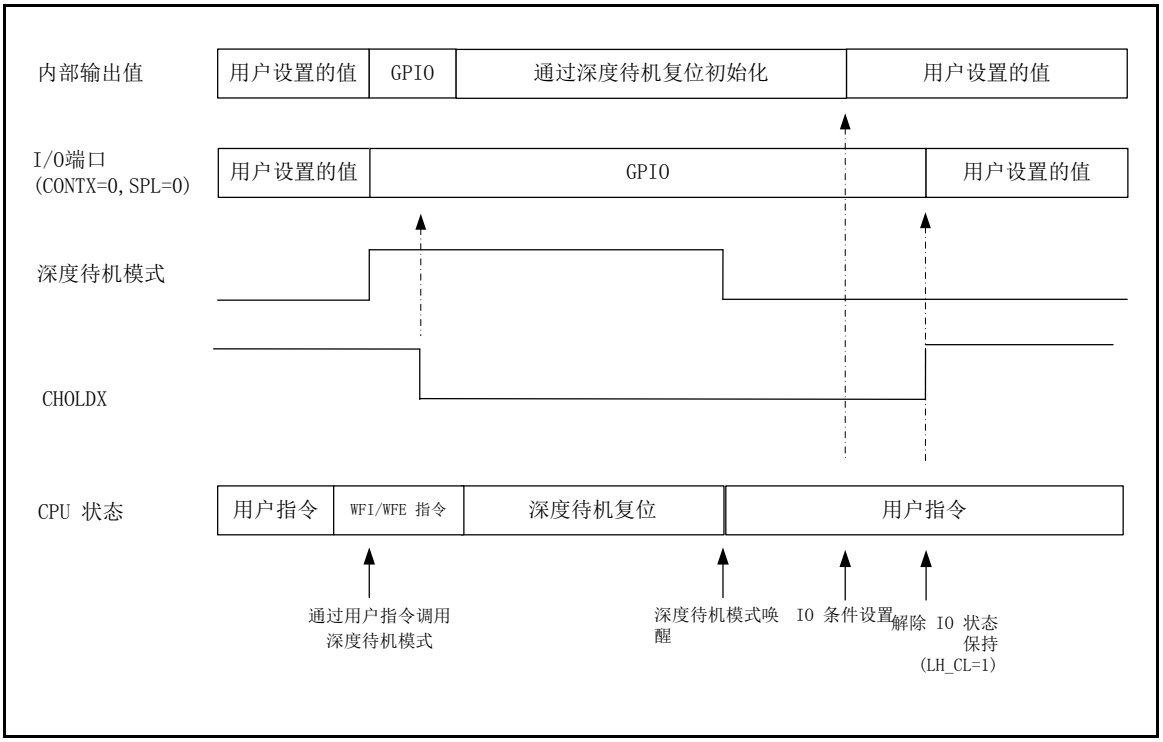
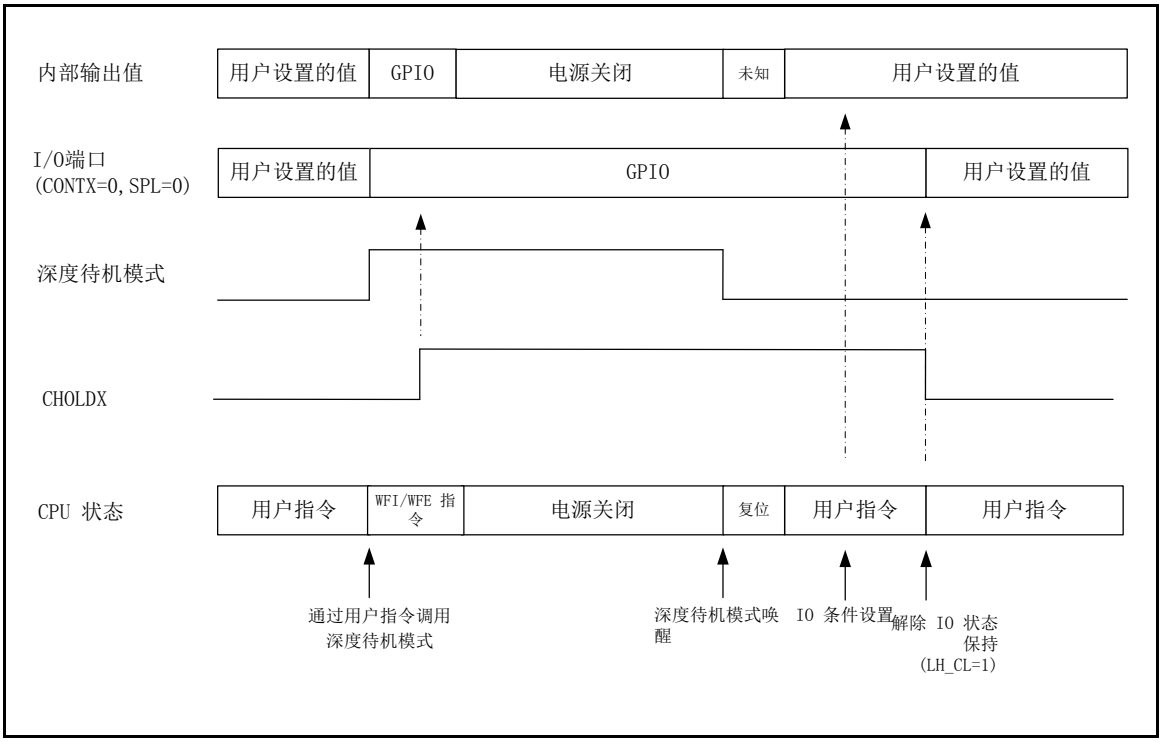


Figure 2-8 所示为 WIOLC_CTL:CONTX=0 时 I/O 端口在深度待机模式时的操作。

Figure 2-8 深度待机模式时的 I/O 端口操作（CONTX=0，TYPE 3-M0+产品）



在本示例中，I/O 端口状态在深度待机模式持续期内为 GPIO 状态。从深度待机模式唤醒后，将保持 I/O 端口状态，直到解除 I/O 锁存。深度待机转换复位不能初始化所有 GPIO 控制寄存器。解除 I/O 锁存之前，CPU 需要进行初始设置以控制 GPIO 寄存器。然后，CPU 将解除 I/O 锁存。这可通过 WIOLC_CTL:LH_CL 写入 1 实现解除。在解除 I/O 锁存之前改变 GPIO 的控制寄存器不会影响 I/O 端口的状态。

有关 WIOLC_CTL 寄存器的详细说明，参见 “6-1 低功耗模式”。

注意事项：

- *部分端口没有选择 GPIO 功能。有关各引脚在深度待机模式下的状态，参见所使用产品《数据手册》中的引脚状态表。*
- *如果使用快速 GPIO 功能，深度待机转换时， I/O 端口状态适用于快速 GPIO 设置而不是 GPIO 设置。*

2.7 深度待机转换复位

Table 2-5 列出深度待机转换复位时是否初始化的寄存器。

Table 2-5 深度待机转换复位

缩写	TYPE 2-M0+	TYPE 3-M0+
PFRx	初始化 *1 *2	初始化 *3
PCRx	不初始化	I/O 端口被锁存后初始化
DDRx	不初始化	I/O 端口被锁存后初始化
PDIRx	不初始化	I/O 端口被锁存后初始化
PDORx	不初始化	I/O 端口被锁存后初始化
ADE	不初始化	I/O 端口被锁存后初始化
EPFRx	不初始化	I/O 端口被锁存后初始化
SPSR	不初始化	I/O 端口被锁存后初始化
PZRx	不初始化	I/O 端口被锁存后初始化
LVDIE	不初始化	不适用
FPDIRx, M_FPDIRx *4	不初始化	I/O 端口被锁存后初始化
FPDORx, M_FPDORx *4	不初始化	I/O 端口被锁存后初始化
FPOERx *4	不初始化	I/O 端口被锁存后初始化

*1: PFR0 寄存器的 bit[4:0] 不初始化。

*2: 以下位在相应条件下不初始化

使用 CEC0_1 时，PFR0 寄存器的 bit11 不初始化。

使用 CEC1_0 时，PFR6 寄存器的 bit0 不初始化。

使用 CEC0_0 时，PFR4 寄存器的 bit12 不初始化。

使用 CEC1_1 时，PFR5 寄存器的 bit6 不初始化。

*3: I/O 端口被锁存后，初始化 PFR0 寄存器的 bit[1]、bit[3]。

I/O 端口被锁存前，初始化除上述寄存器之外的寄存器。

*4: 有关这些寄存器的说明，参见 “快速 GPIO” 一章。

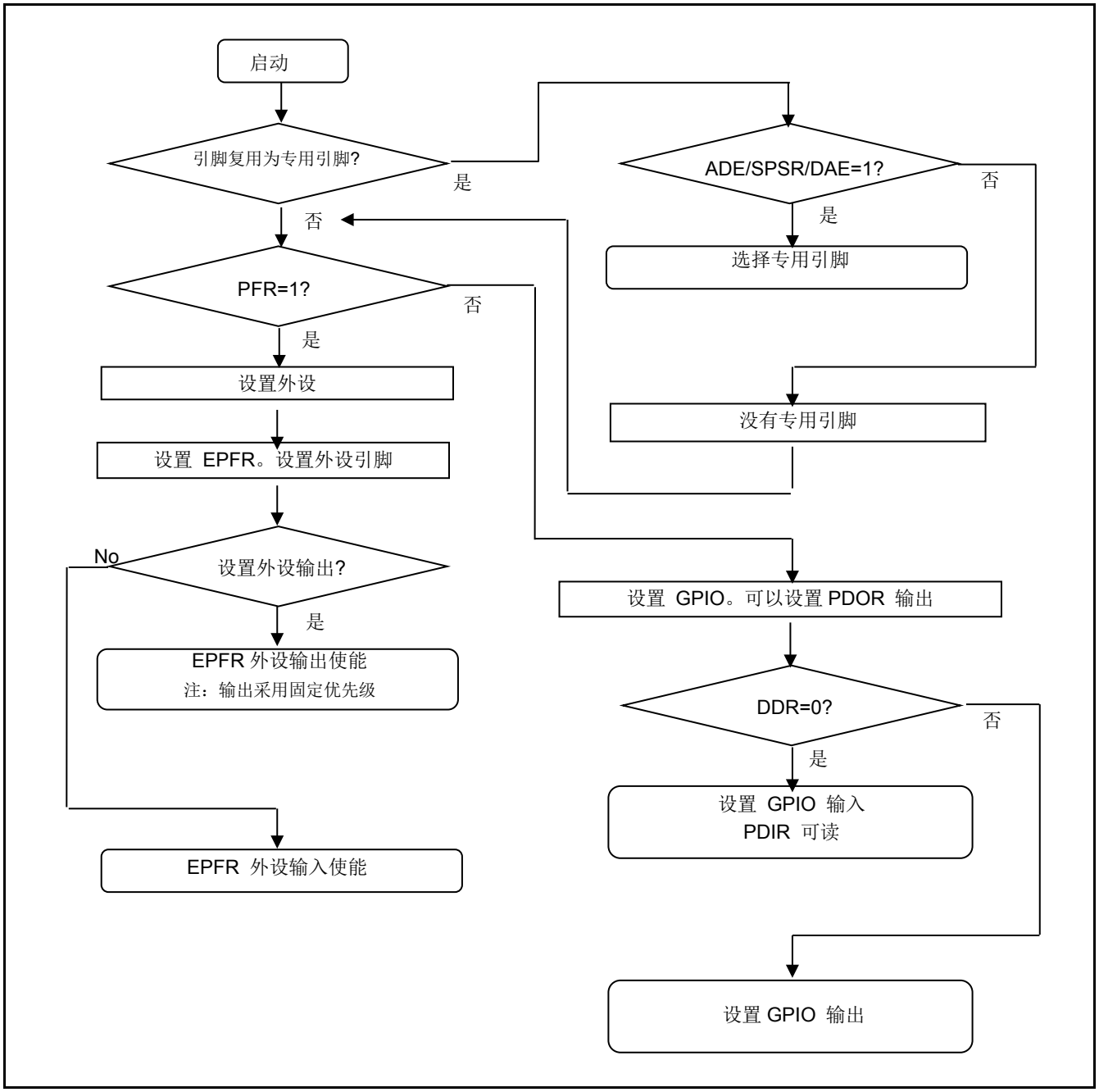
3. 设置步骤示例

本节说明 I/O 端口设置步骤示例。

3.1 I/O 端口设置

设置 I/O 端口寄存器，选择 I/O 方向并选择 GPIO/外设。
Figure 3-1 所示为设置步骤示例。

Figure 3-1 I/O 端口设置步骤示例



3.2 如何使用 I/O 状态保持功能（TYPE 3-M0+产品）

Figure 3-2 所示为如何使能/禁用 I/O 状态保持功能。应在转为深度待机模式前进行本设置。

Figure 3-2I/O 状态保持功能设置步骤示例 1（TYPE 3-M0+产品）

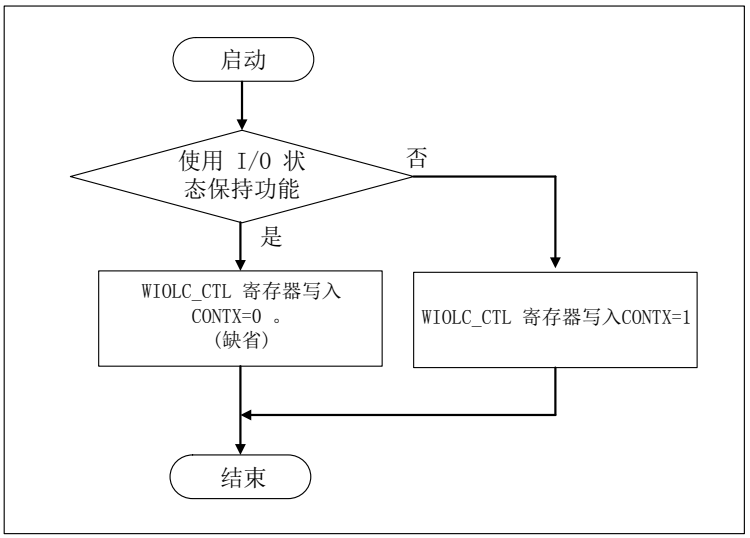
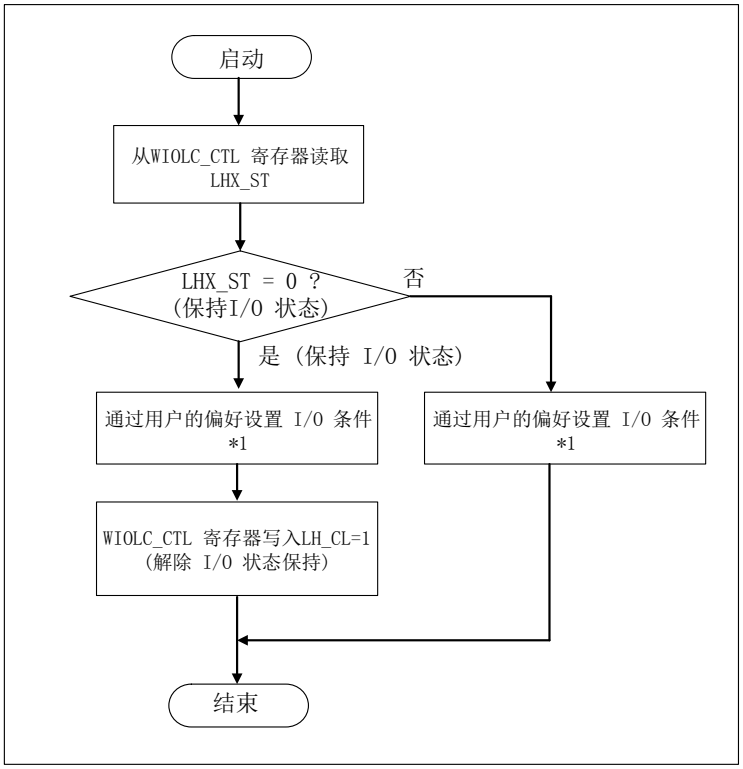


Figure 3-3 所示为如何使用深度待机模式唤醒后的 I/O 状态保持功能。

Figure 3-3I/O 状态保持功能设置步骤示例 2（TYPE 3-M0+产品）



*1： 参见 Figure 3-1 I/O 端口设置步骤示例。

4. 寄存器

本节说明 I/O 端口寄存器表。

Table 4-1 列出寄存器表。

Table 4-1I/O 端口寄存器列表

缩写	寄存器名称	参考章节
PFR0	端口功能设置寄存器 0	4.1
PFR1	端口功能设置寄存器 1	
PFR2	端口功能设置寄存器 2	
PFR3	端口功能设置寄存器 3	
PFR4	端口功能设置寄存器 4	
PFR5	端口功能设置寄存器 5	
PFR6	端口功能设置寄存器 6	
PFR7	端口功能设置寄存器 7	
PFR8	端口功能设置寄存器 8	
PFR9	端口功能设置寄存器 9	
PFRA	端口功能设置寄存器 A	
PFRB	端口功能设置寄存器 B	
PFRC	端口功能设置寄存器 C	
PFRD	端口功能设置寄存器 D	
PFRE	端口功能设置寄存器 E	
PFRF	端口功能设置寄存器 F	
PCR0	上拉设置寄存器 0	4.2
PCR1	上拉设置寄存器 1	
PCR2	上拉设置寄存器 2	
PCR3	上拉设置寄存器 3	
PCR4	上拉设置寄存器 4	
PCR5	上拉设置寄存器 5	
PCR6	上拉设置寄存器 6	
PCR7	上拉设置寄存器 7	
PCR9	上拉设置寄存器 9	
PCRA	上拉设置寄存器 A	
PCRB	上拉设置寄存器 B	
PCRC	上拉设置寄存器 C	
PCRD	上拉设置寄存器 D	
PCRE	上拉设置寄存器 E	
PCRF	上拉设置寄存器 F	

缩写	寄存器名称	参考章节
DDR0	端口输入/输出方向设置寄存器 0	4.3
DDR1	端口输入/输出方向设置寄存器 1	
DDR2	端口输入/输出方向设置寄存器 2	
DDR3	端口输入/输出方向设置寄存器 3	
DDR4	端口输入/输出方向设置寄存器 4	
DDR5	端口输入/输出方向设置寄存器 5	
DDR6	端口输入/输出方向设置寄存器 6	
DDR7	端口输入/输出方向设置寄存器 7	
DDR8	端口输入/输出方向设置寄存器 8	
DDR9	端口输入/输出方向设置寄存器 9	
DDRA	端口输入/输出方向设置寄存器 A	
DDRB	端口输入/输出方向设置寄存器 B	
DDRC	端口输入/输出方向设置寄存器 C	
DDRD	端口输入/输出方向设置寄存器 D	
DDRE	端口输入/输出方向设置寄存器 E	
DDRF	端口输入/输出方向设置寄存器 F	
PDIR0	端口输入数据寄存器 0	4.4
PDIR1	端口输入数据寄存器 1	
PDIR2	端口输入数据寄存器 2	
PDIR3	端口输入数据寄存器 3	
PDIR4	端口输入数据寄存器 4	
PDIR5	端口输入数据寄存器 5	
PDIR6	端口输入数据寄存器 6	
PDIR7	端口输入数据寄存器 7	
PDIR8	端口输入数据寄存器 8	
PDIR9	端口输入数据寄存器 9	
PDIRA	端口输入数据寄存器 A	
PDIRB	端口输入数据寄存器 B	
PDIRC	端口输入数据寄存器 C	
PDIRD	端口输入数据寄存器 D	
PDIRE	端口输入数据寄存器 E	
PDIRF	端口输入数据寄存器 F	
PDOR0	端口输出数据寄存器 0	4.5
PDOR1	端口输出数据寄存器 1	
PDOR2	端口输出数据寄存器 2	
PDOR3	端口输出数据寄存器 3	
PDOR4	端口输出数据寄存器 4	
PDOR5	端口输出数据寄存器 5	
PDOR6	端口输出数据寄存器 6	
PDOR7	端口输出数据寄存器 7	
PDOR8	端口输出数据寄存器 8	
PDOR9	端口输出数据寄存器 9	
PDORA	端口输出数据寄存器 A	
PDORB	端口输出数据寄存器 B	

缩写	寄存器名称	参考章节
PDORC	端口输出数据寄存器 C	
PDORD	端口输出数据寄存器 D	
PDORE	端口输出数据寄存器 E	
PDORF	端口输出数据寄存器 F	
ADE	模拟输入设置寄存器	4.6
EPFR	扩展引脚功能设置寄存器	4.7
EPFR00	扩展引脚功能设置寄存器 00	4.8
EPFR01	扩展引脚功能设置寄存器 01	4.9
EPFR02	扩展引脚功能设置寄存器 02	4.10
EPFR03	扩展引脚功能设置寄存器 03	4.11
EPFR04	扩展引脚功能设置寄存器 04	4.12
EPFR05	扩展引脚功能设置寄存器 05	4.13
EPFR06	扩展引脚功能设置寄存器 06	4.14
EPFR07	扩展引脚功能设置寄存器 07	4.15
EPFR08	扩展引脚功能设置寄存器 08	4.16
EPFR09	扩展引脚功能设置寄存器 09	4.17
EPFR12	扩展引脚功能设置寄存器 12	4.18
EPFR13	扩展引脚功能设置寄存器 13	4.19
EPFR14	扩展引脚功能设置寄存器 14	4.20
EPFR15	扩展引脚功能设置寄存器 15	4.21
EPFR16	扩展引脚功能设置寄存器 16	4.22
EPFR17	扩展引脚功能设置寄存器 17	4.23
EPFR18	扩展引脚功能设置寄存器 18	4.24
EPFR21	扩展引脚功能设置寄存器 21	4.25
EPFR22	扩展引脚功能设置寄存器 22	4.26
EPFR23	扩展引脚功能设置寄存器 23	4.27
EPFR31	扩展引脚功能设置寄存器 31	4.28
EPFR33	扩展引脚功能设置寄存器 33	4.29
EPFR34	扩展引脚功能设置寄存器 34	4.30
EPFR37	扩展引脚功能设置寄存器 37	4.31
EPFR38	扩展引脚功能设置寄存器 38	4.32
SPSR	专用端口设置寄存器	4.33
PZR0	端口伪开漏设置寄存器 0	4.34
PZR1	端口伪开漏设置寄存器 1	
PZR2	端口伪开漏设置寄存器 2	
PZR3	端口伪开漏设置寄存器 3	
PZR4	端口伪开漏设置寄存器 4	
PZR5	端口伪开漏设置寄存器 5	
PZR6	端口伪开漏设置寄存器 6	
PZR7	端口伪开漏设置寄存器 7	
PZR8	端口伪开漏设置寄存器 8	
PZR9	端口伪开漏设置寄存器 9	
PZRA	端口伪开漏设置寄存器 A	
PZRB	端口伪开漏设置寄存器 B	

缩写	寄存器名称	参考章节
PZRC	端口伪开漏设置寄存器 C	
PZRD	端口伪开漏设置寄存器 D	
PZRE	端口伪开漏设置寄存器 E	
PZRF	端口伪开漏设置寄存器 F	
LVDIE	LVDI 输入使能设置寄存器	4.35

4.1 端口功能设置寄存器 (PFRx)

PFRx 寄存器用于选择使用引脚。

PFR 寄存器配置表

位 31 16	15 0	初始值	属性	对应端口
保留	PFR0	0x000A	R/W	P0F 至 P00
保留	PFR1	0x0000	R/W	P1F 至 P10
保留	PFR2	0x0000	R/W	P2F 至 P20
保留	PFR3	0x0000	R/W	P3F 至 P30
保留	PFR4	0x0000	R/W	P4F 至 P40
保留	PFR5	0x0000	R/W	P5F 至 P50
保留	PFR6	0x0000	R/W	P6F 至 P60
保留	PFR7	0x0000	R/W	P7F 至 P70
保留	PFR8	0x0000	R/W	P8F 至 P80
保留	PFR9	0x0000	R/W	P9F 至 P90
保留	PFRA	0x0000	R/W	PAF 至 PA0
保留	PFRB	0x0000	R/W	PBF 至 PB0
保留	PFRC	0x0000	R/W	PCF 至 PC0
保留	PFRD	0x0000	R/W	PDF 至 PD0
保留	PFRE	0x0000	R/W	PEF 至 PE0
保留	PFRF	0x0000	R/W	PFF 至 PF0

寄存器详细配置

位	31 16	15 0
字段	保留	PFRx

寄存器功能

[bit31:16] 保留：保留位

这些位读为 "0x0000"。

写入这些位时，将其设置为 "0x0000"。

[bit15:0] PFRx：端口功能设置寄存器 x

选择使用引脚。

bit15:0		描述
读取		可读取寄存器的设置值。
写入	0	将引脚用作 GPIO 引脚。
	1	将引脚用作外设功能输入/输出引脚。

注意事项：

- *PFRx 中的 "x" 是通配符。 PFRx 指 PFR0、PFR1、PFR2 等。*
- *Px0 和 PxF 中的 "x" 是通配符。 Px0 指 P00, 、P10、P20 等。 PxF 指 P0F、P1F、P2F 等。*
- *可设置 PxF 至 Px0 的 16 个端口功能。*
- *寄存器各位分别设置各引脚。位分配与各引脚序列一一对应。例如，PFR0 的第 15 位设置 P0F，PFR0 的第 14 位设置 P0E，PFR0 的第 0 位设置 P00。*
- *P01 和 P03 被选择为串行线调试引脚时，初始值为 1"。*
- *初始值仅仅是 TYPE 1-M0+ 产品的一个示例，有关不同产品类型，参见附录。*

- *产品中没有的引脚，对应位写入值无效且未定义相应读取值。*

4.2 上拉设置寄存器 (PCRx)

PCRx 寄存器用于设置引脚的上拉。

PCR 寄存器配置表

位 31 16	15 0	初始值	属性	对应端口
保留	PCR0	0x000A	R/W	P0F 至 P00
保留	PCR1	0x0000	R/W	P1F 至 P10
保留	PCR2	0x0000	R/W	P2F 至 P20
保留	PCR3	0x0000	R/W	P3F 至 P30
保留	PCR4	0x0000	R/W	P4F 至 P40
保留	PCR5	0x0000	R/W	P5F 至 P50
保留	PCR6	0x0000	R/W	P6F 至 P60
保留	PCR7	0x0000	R/W	P7F 至 P70
保留	PCR8	0x0000	R/W	P8F 至 P80
保留	PCR9	0x0000	R/W	P9F 至 P90
保留	PCRA	0x0000	R/W	PAF 至 PA0
保留	PCRB	0x0000	R/W	PBF 至 PB0
保留	PCRC	0x0000	R/W	PCF 至 PC0
保留	PCRD	0x0000	R/W	PDF 至 PD0
保留	PCRE	0x0000	R/W	PEF 至 PE0
保留	PCRF	0x0000	R/W	PFF 至 PF0

详细寄存器配置

位	31 16	15 0
字段	保留	PCRx

寄存器功能

[bit31:16] 保留：寄存器位

这些位的读取值为 "0x0000"。

写入这些位时，将其设置为 "0x0000"。

[bit15:0] PCRx：上拉设置寄存器 x

设置引脚的上拉

bit15:0		描述
读取		可读取寄存器的设置值。
写入	0	断开引脚上拉电阻器。
	1	引脚处于输入状态时（GPIO 和外设功能），连接上拉电阻器。 引脚处于输出状态时，断开上拉电阻器。

注意事项:

- *PCRx 中的 "x" 是通配符。 PCRx 指 PCR0、PCR1、PCR2 等。*
- *Px0 和 PxF 中的 "x" 是通配符。Px0 指 P00, 、P10、P20 等。 PxF 指 P0F、P1F、P2F 等。*
- *一个寄存器可设置 PxF 至 Px0 一共 16 个上拉。*
- *寄存器各位分别设置各引脚。位分配与各引脚序列一一对应。例如，PCR0 的第 15 位设置 P0F，PCR0 的第 14 位设置 P0E，PCR0 的第 0 位设置 P00。*
- *P01 和 P03 被选为串行调试引脚时，初始值为 1"。*
- *使用 I^2C 功能时，通过设置 PCRx=0 使用外部上拉。*
- *没有 PCR8。*
- *产品中没有的引脚，对应位写入值无效且未定义相应读取值。*
- *PE0、PE1 端口没有上拉电阻器。因此，写入 PE 寄存器的值无效。读取本寄存器的初始值或写入值。*
- *初始值仅仅是 TYPE 1-M0+ 产品的一个示例，有关不同产品类型，参见附录。*

4.3 端口输入/输出方向设置寄存器 (DDRx)

DDRx 寄存器设置引脚输入/输出方向。

DDR 寄存器配置表

位 31 16	15 0	初始值	属性	对应端口
保留	DDR0	0x0000	R/W	P0F 至 P00
保留	DDR1	0x0000	R/W	P1F 至 P10
保留	DDR2	0x0000	R/W	P2F 至 P20
保留	DDR3	0x0000	R/W	P3F 至 P30
保留	DDR4	0x0000	R/W	P4F 至 P40
保留	DDR5	0x0000	R/W	P5F 至 P50
保留	DDR6	0x0000	R/W	P6F 至 P60
保留	DDR7	0x0000	R/W	P7F 至 P70
保留	DDR8	0x0000	R/W	P8F 至 P80
保留	DDR9	0x0000	R/W	P9F 至 P90
保留	DDRA	0x0000	R/W	PAF 至 PA0
保留	DDRB	0x0000	R/W	PBF 至 PB0
保留	DDRC	0x0000	R/W	PCF 至 PC0
保留	DDRD	0x0000	R/W	PDF 至 PD0
保留	DDRE	0x0000	R/W	PEF 至 PE0
保留	DDRF	0x0000	R/W	PFF 至 PF0

详细寄存器配置

位	31 16	15 0
字段	保留	DDRx

寄存器功能

[bit31:16] 保留：保留位

这些位的读取值为 "0x0000"。

写入这些位时，将其设置为 "0x0000"。

[bit15:0] DDRx：端口输入/输出方向设置寄存器 x

设置引脚输入/输出方向。

bit15:0		描述
读取		可读取寄存器的设置值。
写入	0	在输入方向使用 GPIO。 如果引脚被选为外设功能输入/输出引脚，则设置值无效。
	1	在输出方向使用 GPIO。 如果引脚被选为外设功能输入/输出引脚，则设置值无效。

注意事项:

- *DDRx 中的 "x" 是通配符。 DDRx 指 DDR0、DDR1、DDR2 等。*
- *Px0 和 PxF 中的 "x" 是通配符。 Px0 指 P00, 、P10、P20 等。 PxF 指 P0F、P1F、P2F 等。*
- *寄存器能从 PxF 至 Px0 设置 16 个端口的输入/输出方向。*
- *寄存器各位分别设置各引脚。位分配与各引脚序列一一对应。例如，DDR0 的第 15 位设置 P0F，DDR0 的第 14 位设置 P0E，DDR0 的第 0 位设置 P00。*
- *如果选择多功能计时器的输出 RTO，由于 DTTIX 信号而紧急停止时，DDR 将控制引脚状态。有关详细说明，参见《计时器部分》中的“多功能计时器”一章。*
- *产品中没有的引脚，对应位写入值无效且未定义相应读取值。*
- *初始值仅仅是 TYPE 1 产品的一个示例，有关不同产品类型，参见附录。*

4.4 端口输入数据寄存器 (PDIRx)

PDIRx 寄存器用于指示引脚的输入数据。

PDIR 寄存器配置表

位 31 16	15 0	初始值	属性	对应端口
保留	PDIR0	0xXXXX	R	P0F 至 P00
保留	PDIR1	0xXXXX	R	P1F 至 P10
保留	PDIR2	0xXXXX	R	P2F 至 P20
保留	PDIR3	0xXXXX	R	P3F 至 P30
保留	PDIR4	0xXXXX	R	P4F 至 P40
保留	PDIR5	0xXXXX	R	P5F 至 P50
保留	PDIR6	0xXXXX	R	P6F 至 P60
保留	PDIR7	0xXXXX	R	P7F 至 P70
保留	PDIR8	0xXXXX	R	P8F 至 P80
保留	PDIR9	0xXXXX	R	P9F 至 P90
保留	PDIRA	0xXXXX	R	PAF 至 PA0
保留	PDIRB	0xXXXX	R	PBF 至 PB0
保留	PDIRC	0xXXXX	R	PCF 至 PC0
保留	PDIRD	0xXXXX	R	PDF 至 PD0
保留	PDIRE	0xXXXX	R	PEF 至 PE0
保留	PDIRF	0xXXXX	R	PFF 至 PF0

详细寄存器配置

位	31 16	15 0
字段	保留	PDIRx

寄存器功能

[bit31:16] 保留：保留位

这些位的读取值为 "0x0000"。

写入这些位时，将其设置为 "0x0000"。

[bit15:0] PDIRx：端口输入数据寄存器 x

读取引脚的输入数据。

bit15:0		描述
读取	0	不管引脚功能如何设置 (PFR/EPFR/DDR/PDOR)，此位指示引脚处于 "L" 电平输入或 "L" 电平输出状态。通过 ADE/SPSR 选择专用引脚时，由于输入被切断，总是读出 0。
	1	不管引脚功能如何设置 (PFR/EPFR/DDR/PDOR)，此位指示引脚处于 "H" 电平输入或 "H" 电平输出状态。
写入		写入无效。

注意事项:

- *PDIRx 中的 "x" 是通配符。 PDIRx 指 PDIR0、PDIR1、PDIR2 等。*
- *Px0 和 PxF 中的 "x" 是通配符。 Px0 指 P00, 、P10、P20 等。 PxF 指 P0F、P1F、P2F 等。*
- *一个寄存存器可读取 PxF 至 Px0 共 16 个端口的输入数据。*
- *寄存器各位分别指示各个引脚的状态。位分配与各引脚序列一一对应。例如，PDIR0 的第 15 位指示 P0F，PDIR0 的第 14 位指示 P0E，PDIR0 的第 0 位指示 P00。*
- *产品没有配置的引脚对应位的值总是读为 "0"。*
- *初始值仅仅是 TYPE 1-M0+ 产品的一个示例，有关不同产品类型，参见附录。*

4.5 端口输出数据寄存器 x (PDORx)

PDORx 寄存器用于设置引脚的输出数据。

PDOR 寄存器配置表

位 31 16	15 0	初始值	属性	对应端口
保留	PDOR0	0x0000	R/W	P0F 至 P00
保留	PDOR1	0x0000	R/W	P1F 至 P10
保留	PDOR2	0x0000	R/W	P2F 至 P20
保留	PDOR3	0x0000	R/W	P3F 至 P30
保留	PDOR4	0x0000	R/W	P4F 至 P40
保留	PDOR5	0x0000	R/W	P5F 至 P50
保留	PDOR6	0x0000	R/W	P6F 至 P60
保留	PDOR7	0x0000	R/W	P7F 至 P70
保留	PDOR8	0x0000	R/W	P8F 至 P80
保留	PDOR9	0x0000	R/W	P9F 至 P90
保留	PDORA	0x0000	R/W	PAF 至 PA0
保留	PDORB	0x0000	R/W	PBF 至 PB0
保留	PDORC	0x0000	R/W	PCF 至 PC0
保留	PDORD	0x0000	R/W	PDF 至 PD0
保留	PDORE	0x0000	R/W	PEF 至 PE0
保留	PDORF	0x0000	R/W	PFF 至 PF0

详细寄存器配置

位	31 16	15 0
字段	保留	PDORx

寄存器功能

[bit31:16] 保留：保留位

这些位的读取值为 "0x0000"。

写入这些位时，将其设置为 "0x0000"。

[bit15:0] PDORx：端口输出数据寄存器 x

设置引脚输出数据。

bit15:0		描述
读取		读取寄存器值。
写入	0	向 GPIO 输出 “L” 电平。 如果引脚被选为 I/O 输入或外设功能输入/输出，则设置值无效。
	1	向 GPIO 输出 “H” 电平。 如果引脚被选为 I/O 输入或外设功能输入/输出，则设置值无效。

注意事项:

- *PDORx 中的 "x" 是通配符。 PDORx 指 PDOR0、PDOR1、PDOR2 等。*
- *Px0 和 PxF 中的 "x" 是通配符。 Px0 指 P00, 、P10、P20 等。 PxF 指 P0F、P1F、P2F 等。*
- *一个寄存器可设置 PxF 至 Px0 共 16 个端口的输出数据。*
- *寄存器各位分别设置各引脚。位分配与各引脚序列一一对应。例如，PDOR0 的第 15 位设置 P0F，PDOR0 的第 14 位设置 P0E，PDOR0 的第 0 位设置 P00。*
- *产品中没有的引脚，对应位写入值无效且未定义相应读取值。*
- *初始值仅仅是 TYPE 1-M0+ 产品的一个示例，有关不同产品类型，参见附录。*

4.6 模拟输入设置寄存器 (ADE)

ADE 寄存器将外部引脚设置为 ADC 的模拟信号输入引脚。

寄存器配置

位	31															0
字段	ADE															
属性	R/W															
初始值	0xFFFFFFFF															

寄存器功能

[bit31:0] ADE：模拟输入设置寄存器

设置为模拟信号输入引脚。

bit31:0		描述
读取		读取寄存器值。
写入	0	将外部引脚用作数字输入/输出而不是模拟输入。
	1	将外部引脚用作模拟输入。 （I/O 单元将处于输入方向状态、输入切断状态和上拉断开状态。）

注意事项：

- *本寄存器设置 AN31 至 AN00 的模拟输入引脚。*
- *寄存器各位分别设置各引脚。位分配与各引脚序列一一对应。例如，ADE 的第 31 位设置 AN31，ADE 的第 14 位设置 AN14，ADE 的第 0 位设置 AN00。不同产品中，ANxx 端口位置各不相同。*
  有关对应关系，参见所使用产品的《数据手册》。
- *若为 WKUPx 引脚和 ANx 引脚复用的端口，禁止设置成 (WIER:WUEx=1) & (ADEx=1))。*

4.7 扩展引脚功能设置寄存器 (EPFRx)

如果不止一种功能时，采用 EPFRx 寄存器分配引脚的功能。

EPFRx 寄存器配置表

位 31 0	初始值	属性	对应功能
EPFR00	0x00010000	R/W	系统功能
EPFR01	0x00000000	R/W	多功能计时器
EPFR02	0x00000000	R/W	
EPFR03	0x00000000	R/W	
EPFR04	0x00000000	R/W	基本计时器
EPFR05	0x00000000	R/W	
EPFR06	0x00000000	R/W	外部中断
EPFR07	0x00000000	R/W	多功能串口
EPFR08	0x00000000	R/W	
EPFR09	0x00000000	R/W	CAN/ADC 触发信号 / QPRC
EPFR12	0x00000000	R/W	基本计时器
EPFR13	0x00000000	R/W	
EPFR14	0x00000000	R/W	QPRC
EPFR15	0x00000000	R/W	外部中断
EPFR16	0x00000000	R/W	多功能串口
EPFR17	0x00000000	R/W	
EPFR18	0x00000000	R/W	HDMI-CEC / 远控接收
EPFR21	0x00000000	R/W	QPRC
EPFR22	0x00000000	R/W	多功能串口
EPFR23	0x00000000	R/W	多功能串口
EPFR31	0x00000000	R/W	I2C 从机唤醒
EPFR33	0x00000000	R/W	IC 卡
EPFR34	0x00000000	R/W	多功能串口。流控制
EPFR37	0x00000000	R/W	多功能串口 I2S
EPFR38	0x00000000	R/W	多功能串口 I2S

各类型产品配备不同的 EPFRx 寄存器。

有关 EPFRx 寄存器和产品类型之间的对应关系，参见 Table 4-2。

注意事项:

- *初始值仅仅是 TYPE 1-M0+ 产品的一个示例，有关不同产品类型，参见附录。*

Table 4-2EPFRx 寄存器产品类型对应表

	TYPE 1-M0+	TYPE 2-M0+	TYPE 3-M0+
EPFR00	□	□	□
EPFR01	□	□	-
EPFR02	-	-	-
EPFR03	-	-	-
EPFR04	□	□	□
EPFR05	-	□	□
EPFR06	□	□	□
EPFR07	□	□	□
EPFR08	-	□	□
EPFR09	□	□	□
EPFR12	-	-	-
EPFR13	-	-	-
EPFR14	-	-	-
EPFR15	-	□	-
EPFR16	-	□	-
EPFR17	-	-	-
EPFR18	-	□	-
EPFR21	□	-	-
EPFR22	□	-	□
EPFR23	-	□	□
EPFR31	-	-	□
EPFR33	-	□	□
EPFR34	-	□	-
EPFR37	-	□	□
EPFR38	-	□	□

4.8 扩展引脚功能设置寄存器 00(EPFR00)

如果不止一种功能时，采用 EPFR00 寄存器分配引脚的功能。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	保留							
属性				-				
初始值				-				

位	23	22	21	20	19	18	17	16
字段	保留							SWDEN
属性				-				R/W
初始值				-				1

位	15	14	13	12	11	10	9	8
字段	保留						USBP0E	保留
属性				-			R/W	-
初始值				-			0	-

位	7	6	5	4	3	2	1	0
字段	SUBOUTE		RTCCOE		保留	CROUTE		NMIS
属性	R/W		R/W		-	R/W		R/W
初始值	00		00		-	00		0

寄存器功能

[bit31:17] 保留：保留位

这些位的读取值为 "0"。

写入这些位时，将其设置为 "0"。

[bit16] SWDEN：串行线调试功能选择位 0

选择 SWCLK 和 SWDIO 引脚功能。

位		描述
读取		读取寄存器值。
写入	0	不使用 SWCLK 和 SWDIO 两个引脚。 （可使用复用引脚。）
	1	使用 SWCLK 和 SWDIO 两个引脚。[初始值]

注意事项：

- *使能 SWD 时，必须将 DEBUG_SW_CTL 寄存器的 DBG_EN 位设置为 1，以进一步使能 SWDEN。*

[bit15:10] 保留：保留位
这些位的读取值为 "0"。
写入这些位时，将其设置为 "0"。

[bit9] USBP0E： USBch.0 功能选择位 1
选择 USB ch.0 功能。

位		描述
读取		读取寄存器值。
写入	0	USBch.0 不产生输出 D+ 电阻控制信号 (HCONTX)。[初始值] （可使用复用引脚。）
	1	产生 USBch.0 的输出 D+ 电阻控制信号 (HCONTX)。

[bit8:0] 保留：保留位
这些位的读取值为 "0"。
写入这些位时，将其设置为 "0"。

[bit7:6] SUBOUTE：副时钟分频输出功能选择位
选择副时钟分频输出。

bit7:6		描述
读取		读取寄存器值。
写入	00	不执行副时钟分频输出。[初始值]
	01	SUBOUT_0 用作副时钟分频输出引脚。
	10	SUBOUT_1 用作副时钟分频输出引脚。
	11	SUBOUT_2 用作副时钟分频输出引脚。

[bit5:4] RTCCOE：RTC 时钟输出选择位
选择 RTC 时钟输出。

bit5:4		描述
读取		读取寄存器值。
写入	00	不执行 RTC 时钟输出。[初始值]
	01	RTCCOE_0 用作 RTC 时钟输出引脚。
	10	RTCCOE_1 用作 RTC 时钟输出引脚。
	11	RTCCOE_2 用作 RTC 时钟输出引脚。

[bit3] 保留：保留位
此位读取值为"0"。
写入此位时，将其设置为 "0"。

[bit2:1] CROUTE：内部高速 CR 振荡输出功能选择位
选择内部高速 CR 振荡输出。

bit2:1		描述
读取		读取寄存器值。
写入	00	不产生内部高速 CR 振荡输出。[初始值]
	01	内部高速 CR 振荡输出引脚使用 CROUT_0。
	10	内部高速 CR 振荡输出引脚使用 CROUT_1。

	11	内部高速 CR 振荡输出引脚使用 CROUT_2。

[bit0] NMIS：NMIX 功能选择位

选择 NMIX 引脚功能。

位		描述
读取		读取寄存器值。
写入	0	不使用 NMIX 引脚。[初始值]
	1	使用 NMIX 引脚。

注意事项：

- *使用 NMIX 引脚时，设置 NMIS="1" 和 PFR="1"。*

若为 TYPE 1-M0+、TYPE 2-M0+ 产品

- *当映射至 NMI 输入引脚的 I/O 端口改变为从 NMI 至 GPIO 或其他外设功能（写入 EPFR00.NMIS = 1）时，I/O 端口的输入电平应保持高电平并改变 I/O 端口。若 I/O 端口被选为 GPIO 或其他外设功能，内部 NMI 信号将受高电平约束。因此，当 I/O 端口的输入电平为低电平时，I/O 端口将从 GPIO 或其他外设功能变为 NMI，内部 NMI 信号将从高变低。于是检测到下降沿，产生 NMI 请求。*

若为 TYPE 3-M0+ 产品

- *当映射至 NMI 输入引脚的 I/O 端口改变为从 NMI 至 GPIO 或其他外设功能（写入 EPFR00.NMIS = 1）时，应改变为通过非屏蔽中断使能寄存器 (NMIENR=0) 禁用 NMI 的状态。。改变 GPIO 之后，通过 NMICL 清除 NMIRR。同时通过(NMIENR=1) 使能 NMI。有关详细说明，参见“外部中断及 NMI 控制段”一章。*

4.9 扩展引脚功能设置寄存器 01 (EPFR01)

EPFR01 寄存器将功能分配给多功能计时器单元 0 的引脚。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	IC03S			IC02S			IC01S	
属性	R/W			R/W			R/W	
初始值	000			000			00	

位	23	22	21	20	19	18	17	16
字段	IC01S	IC00S			FRCK0S		DTTI0S	
属性	R/W	R/W			R/W		R/W	
初始值	0	000			00		00	

位	15	14	13	12	11	10	9	8
字段	保留		IGTRG0	DTTI0C	RTO05E		RTO04E	
属性	-		R/W	R/W	R/W		R/W	
初始值	-		0	0	00		00	

位	7	6	5	4	3	2	1	0
字段	RTO03E		RTO02E		RTO01E		RTO00E	
属性	R/W		R/W		R/W		R/W	
初始值	00		00		00		00	

寄存器功能

[bit31:29] IC03S：IC03 输入选择位

选择 IC03 输入。

bit31:29		描述
读取		读取寄存器值。
写入	000	输入捕捉 IC03 的输入引脚使用 IC03_0。[初始值]
	001	与写入 000 时相同。
	010	输入捕捉 IC03 的输入引脚使用 IC03_1。
	011	输入捕捉 IC03 的输入引脚使用 IC03_2。
	100	输入捕捉 IC03 输入采用内部宏 MFSch.3LSYN。
	101	输入捕捉 IC03 输入采用内部宏 MFSch.7LSYN。
	110	禁止设置。
	111	禁止设置。

[bit28:26] IC02S：IC02 输入选择位

选择 IC02 输入。

bit28:26		描述
读取		读取寄存器值。
写入	000	输入捕捉 IC02 的输入引脚使用 IC02_0。[初始值]
	001	与写入 000 时相同。
	010	输入捕捉 IC02 的输入引脚使用 IC02_1。
	011	输入捕捉 IC02 的输入引脚使用 IC02_2。
	100	输入捕捉 IC02 输入采用内部宏 MFSch.2LSYN。
	101	输入捕捉 IC02 输入采用内部宏 MFSch.6LSYN。
	110	禁止设置。
	111	禁止设置。

[bit25:23] IC01S：IC01 输入选择位

选择 IC01 输入。

bit25:23		描述
读取		读取寄存器值。
写入	000	输入捕捉 IC01 的输入引脚使用 IC01_0。[初始值]
	001	与写入 000 时相同。
	010	输入捕捉 IC01 的输入引脚使用 IC01_1。
	011	输入捕捉 IC01 的输入引脚使用 IC01_2。
	100	输入捕捉 IC01 的输入采用内部宏 MFSch.1LSYN。
	101	输入捕捉 IC01 的输入采用内部宏 MFSch.5LSYN。
	110	禁止设置。
	111	禁止设置。

[bit22:20] IC00S：IC00 输入选择位

选择 IC00 输入。

bit22:20		描述
读取		读取寄存器值。
写入	000	输入捕捉 IC00 的输入引脚使用 IC00_0。[初始值]
	001	与写入 000 时相同。
	010	输入捕捉 IC00 的输入引脚使用 IC00_1。
	011	输入捕捉 IC00 的输入引脚使用 IC00_2。
	100	输入捕捉 IC00 的输入采用内部宏 MFSch.0LSYN。
	101	输入捕捉 IC00 的输入采用内部宏 MFSch.4LSYN。
	110	禁止设置。
	111	禁止设置。

[bit19:18] FRCK0S：FRCK0 输入选择位

选择 FRCK0 输入。

bit19:18		描述
读取		读取寄存器值。
写入	00	自由运行计时器 FRCK0 的输入引脚使用 FRCK0_0。[初始值]
	01	与写入 00 时相同。
	10	自由运行计时器 FRCK0 的输入引脚使用 FRCK0_1。
	11	自由运行计时器 FRCK0 的输入引脚使用 FRCK0_2。

[bit17:16] DTTI0S：DTTI0X 输入选择位

选择 DTTI0X 输入。

bit17:16		描述
读取		读取寄存器值。
写入	00	波形发生器 DTTI0X 的输入引脚使用 DTTI0X_0。[初始值]
	01	与写入 00 时相同。
	10	波形发生器 DTTI0X 的输入引脚使用 DTTI0X_1。
	11	波形发生器 DTTI0X 的输入引脚使用 DTTI0X_2。

[bit15:14] 保留：保留位

这些位的读取值为 "0b00"。

写入这些位时，将其设置为 "0b00"。

[bit13] IGTRG0：IGTRG0 输入选择位

选择 IGTRG0 输入。

bit13		描述
读取		读取寄存器值。
写入	0	PPG IGTRG 输入引脚使用 IGTRG0_0。[初始值]
	1	PPG IGTRG 输入引脚使用 IGTRG0_1。

[bit12] DTTI0C： DTTI0X 功能选择位

选择 DTTI0X 功能。

bit12		描述
读取		读取寄存器值。
写入	0	从 RTO00 至 RTO05 引脚输出，不用 DTTIF0 转换 GPIO。[初始值]
	1	DTTIF0 转换 GPIO，从 RTO00 至 RTO05 引脚输出。

[bit11:10] RTO05E：RTO05 输出选择位

选择 RTO05 输出。

bit11:10		描述
读取		读取寄存器值。
写入	00	不产生波形发生器 RTO05 的输出。[初始值]
	01	波形发生器 RTO05 的输出引脚使用 RTO05_0。
	10	波形发生器 RTO05 的输出引脚使用 RTO05_1。
	11	禁止设置。

[bit9:8] RTO04E：RTO04 输出选择位

选择 RTO04 输出。

bit9:8		描述
读取		读取寄存器值。
写入	00	不产生波形发生器 RTO04 的输出。[初始值]
	01	波形发生器 RTO04 的输出引脚使用 RTO04_0。
	10	波形发生器 RTO04 的输出引脚使用 RTO04_1。
	11	禁止设置。

[bit7:6] RTO03E：RTO03 输出选择位

选择 RTO03 输出。

bit7:6		描述
读取		读取寄存器值。
写入	00	不产生波形发生器 RTO03 的输出。[初始值]
	01	波形发生器 RTO03 的输出引脚使用 RTO03_0。
	10	波形发生器 RTO03 的输出引脚使用 RTO03_1。
	11	禁止设置。

[bit5:4] RTO02E：RTO02 输出选择位

选择 RTO02 输出。

bit5:4		描述
读取		读取寄存器值。
写入	00	不产生波形发生器 RTO02 的输出。[初始值]
	01	波形发生器 RTO02 的输出引脚使用 RTO02_0。
	10	波形发生器 RTO02 的输出引脚使用 RTO02_1。
	11	禁止设置。

[bit3:2] RTO01E：RTO01 输出选择位

选择 RTO01 输出。

bit3:2		描述
读取		读取寄存器值。
写入	00	不产生波形发生器 RTO01 的输出。[初始值]
	01	波形发生器 RTO01 的输出引脚使用 RTO01_0。
	10	波形发生器 RTO01 的输出引脚使用 RTO01_1。
	11	禁止设置。

[bit1:0] RTO00E：RTO00 输出选择位

选择 RTO00 输出。

bit1:0		描述
读取		读取寄存器值。
写入	00	不产生波形发生器 RTO00 的输出。[初始值]
	01	波形发生器 RTO00 的输出引脚使用 RTO00_0。
	10	波形发生器 RTO00 的输出引脚使用 RTO00_1。
	11	禁止设置。

4.10 扩展引脚功能设置寄存器 02 (EPFR02)

EPFR02 寄存器分配多功能计时器单元 1 引脚的功能。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	IC13S			IC12S			IC11S	
属性		R/W			R/W			R/W
初始值		000			000			00

位	23	22	21	20	19	18	17	16
字段	IC11S	IC10S			FRCK1S		DTTI1S	
属性	R/W		R/W		R/W		R/W	
初始值	0		000		00		00	

位	15	14	13	12	11	10	9	8
字段	保留			DTTI1C	RTO15E		RTO14E	
属性		-		R/W	R/W		R/W	
初始值		-		0	00		00	

位	7	6	5	4	3	2	1	0
字段	RTO13E		RTO12E		RTO11E		RTO10E	
属性		R/W		R/W		R/W		R/W
初始值		00		00		00		00

寄存器功能

[bit31:29] IC13S：IC13 输入选择位

选择 IC13 输入。

bit31:29		描述
读取		读取寄存器值。
写入	000	输入捕捉 IC13 的输入引脚使用 IC13_0。[初始值]
	001	与写入 000 时相同。
	010	输入捕捉 IC13 的输入引脚使用 IC13_1。
	011	禁止设置。
	100	输入捕捉 IC13 的输入采用内部宏 MFSch.3LSYN。
	101	输入捕捉 IC13 的输入采用内部宏 MFSch.7LSYN。
	110	禁止设置。
	111	禁止设置。

[bit28:26] IC12S：IC12 输入选择位

选择 IC12 输入。

bit28:26		描述
读取		读取寄存器值。
写入	000	输入捕捉 IC12 的输入引脚使用 IC12_0。[初始值]
	001	与写入 000 时相同。
	010	输入捕捉 IC12 的输入引脚使用 IC12_1。
	011	禁止设置。
	100	输入捕捉 IC12 的输入采用内部宏 MFSch.2LSYN。
	101	输入捕捉 IC12 的输入采用内部宏 MFSch.6LSYN。
	110	禁止设置。
	111	禁止设置。

[bit25:23] IC11S：IC11 输入选择位

选择 IC11 输入。

bit25:23		描述
读取		读取寄存器值。
写入	000	输入捕捉 IC11 的输入引脚使用 IC11_0。[初始值]
	001	与写入 000 时相同。
	010	输入捕捉 IC11 的输入引脚使用 IC11_1。
	011	禁止设置。
	100	输入捕捉 IC11 的输入采用内部宏 MFSch.1LSYN。
	101	输入捕捉 IC11 的输入采用内部宏 MFSch.5LSYN。
	110	禁止设置。
	111	禁止设置。

[bit22:20] IC10S：IC10 输入选择位

选择 IC10 输入。

bit22:20		描述
读取		读取寄存器值。
写入	000	输入捕捉 IC10 的输入引脚使用 IC10_0。[初始值]
	001	与写入 000 时相同。
	010	输入捕捉 IC10 的输入引脚使用 IC10_1。
	011	禁止设置。
	100	输入捕捉 IC10 的输入采用内部宏 MFSch.0LSYN。
	101	输入捕捉 IC10 的输入采用内部宏 MFSch.4LSYN。
	110	禁止设置。
	111	禁止设置。

[bit19:18] FRCK1S：FRCK1 输入选择位

选择 FRCK1 输入。

bit19:18		描述
读取		读取寄存器值。
写入	00	自由运行计时器 FRCK1 的输入引脚使用 FRCK1_0。[初始值]
	01	与写入 00 时相同。
	10	自由运行计时器 FRCK1 的输入引脚使用 FRCK1_1。
	11	禁止设置。

[bit17:16] DTTI1S：DTTI1X 输入选择位

选择 DTTI1X 输入。

bit17:16		描述
读取		读取寄存器值。
写入	00	波形发生器 DTTI1X 的输入引脚使用 DTTI1X_0。[初始值]
	01	与写入 00 时相同。
	10	波形发生器 DTTI1X 的输入引脚使用 DTTI1X_1。
	11	禁止设置。

[bit15:13] 保留：保留位

这些位的读取值为 "0b000"。

写入这些位时，将其设置为 "0b000"。

[bit12] DTTI1C： DTTI1X 功能选择位

选择 DTTI1X 功能。

位		描述
读取		读取寄存器值。
写入	0	从 RTO10 至 RTO15 引脚输出，不用 DTTIF1 转换 GPIO。[初始值]
	1	DTTIF1 转换 GPIO，从 RTO10 至 RTO15 引脚输出。

[bit11:10] RTO15E：RTO15 输出选择位

选择 RTO15 输出。

bit11:10		描述
读取		读取寄存器值。
写入	00	不产生波形发生器 RTO15 的输出。[初始值]
	01	波形发生器 RTO15 的输出引脚使用 RTO15_0。
	10	波形发生器 RTO15 的输出引脚使用 RTO15_1。
	11	禁止设置。

[bit9:8] RTO14E：RTO14 输出选择位

选择 RTO14 输出。

bit9:8		描述
读取		读取寄存器值。
写入	00	不产生波形发生器 RTO14 的输出。[初始值]
	01	波形发生器 RTO14 的输出引脚使用 RTO14_0。
	10	波形发生器 RTO14 的输出引脚使用 RTO14_1。
	11	禁止设置。

[bit7:6] RTO13E：RTO13 输出选择位

选择 RTO13 输出。

bit7:6		描述
读取		读取寄存器值。
写入	00	不产生波形发生器 RTO13 的输出。[初始值]
	01	波形发生器 RTO13 的输出引脚使用 RTO13_0。
	10	波形发生器 RTO13 的输出引脚使用 RTO13_1。
	11	禁止设置。

[bit5:4] RTO12E：RTO12 输出选择位

选择 RTO12 输出。

bit5:4		描述
读取		读取寄存器值。
写入	00	不产生波形发生器 RTO12 的输出。[初始值]
	01	波形发生器 RTO12 的输出引脚使用 RTO12_0。
	10	波形发生器 RTO12 的输出引脚使用 RTO12_1。
	11	禁止设置。

[bit3:2] RTO11E：RTO11 输出选择位

选择 RTO11 输出。

bit3:2		描述
读取		读取寄存器值。
写入	00	不产生波形发生器 RTO11 的输出。[初始值]
	01	波形发生器 RTO11 的输出引脚使用 RTO11_0。
	10	波形发生器 RTO11 的输出引脚使用 RTO11_1。
	11	禁止设置。

[bit1:0] RTO10E：RTO10 输出选择位

选择 RTO10 输出。

bit1:0		描述
读取		读取寄存器值。
写入	00	不产生波形发生器 RTO10 的输出。[初始值]
	01	波形发生器 RTO10 的输出引脚使用 RTO10_0。
	10	波形发生器 RTO10 的输出引脚使用 RTO10_1。
	11	禁止设置。

4.11 扩展引脚功能设置寄存器 03 (EPFR03)

EPFR03 寄存器分配多功能计时器单元 2 引脚的功能。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	IC23S			IC22S			IC21S	
属性		R/W			R/W			R/W
初始值		000			000			00

位	23	22	21	20	19	18	17	16
字段	IC21S	IC20S			FRCK2S		DTTI2S	
属性	R/W		R/W		R/W		R/W	
初始值	0		000		00		00	

位	15	14	13	12	11	10	9	8
字段	保留			DTTI2C	RTO25E		RTO24E	
属性		-		R/W	R/W		R/W	
初始值		-		0	00		00	

位	7	6	5	4	3	2	1	0
字段	RTO23E		RTO22E		RTO21E		RTO20E	
属性	R/W		R/W		R/W		R/W	
初始值	00		00		00		00	

寄存器功能

[bit31:29] IC23S：IC23 输入选择位

选择 IC23 输入。

bit31:29		描述
读取		读取寄存器值。
写入	000	IC23_0 用作输入捕捉 IC23 的输入引脚。[初始值]
	001	与写入 "000" 时相同。
	010	IC23_1 用作输入捕捉 IC23 的输入引脚。
	011	禁止设置。
	100	输入捕捉 IC23 的输入采用内部宏 MFSch.3LSYN。
	101	输入捕捉 IC23 的输入采用内部宏 MFSch.7 LSYN。
	110	禁止设置。
	111	禁止设置。

[bit28:26] IC22S：IC22 输入选择位

选择 IC22 输入。

bit28:26		描述
读取		读取寄存器值。
写入	000	IC22_0 用作输入捕捉 IC22 的输入引脚。[初始值]
	001	与写入 "000" 时相同。
	010	IC22_1 用作输入捕捉 IC22 的输入引脚。
	011	禁止设置。
	100	输入捕捉 IC22 的输入采用内部宏 MFSch.2 LSYN。
	101	输入捕捉 IC22 的输入采用内部宏 MFSch.6 LSYN。
	110	禁止设置。
	111	禁止设置。

[bit25:23] IC21S：IC21 输入选择位

选择 IC21 输入。

bit25:23		描述
读取		读取寄存器值。
写入	000	IC21_0 用作输入捕捉 IC21 的输入引脚。[初始值]
	001	与写入 "000" 时相同。
	010	IC21_1 用作输入捕捉 IC21 的输入引脚。
	011	禁止设置。
	100	输入捕捉 IC21 的输入采用内部宏 MFSch.1 LSYN。
	101	输入捕捉 IC21 的输入采用内部宏 MFSch.5 LSYN。
	110	禁止设置。
	111	禁止设置。

[bit22:20] IC20S：IC20 输入选择位

选择 IC20 输入。

bit22:20		描述
读取		读取寄存器值。
写入	000	IC20_0 用作输入捕捉 IC20 的输入引脚。[初始值]
	001	与写入 "000" 时相同。
	010	IC20_1 用作输入捕捉 IC20 的输入引脚。
	011	禁止设置。
	100	输入捕捉 IC20 的输入采用内部宏 MFSch.0 LSYN。
	101	输入捕捉 IC20 的输入采用内部宏 MFSch.4 LSYN。
	110	禁止设置。
	111	禁止设置。

[bit19:18] FRCK2S：FRCK2 输入选择位

选择 FRCK2 输入。

bit19:18		描述
读取		读取寄存器值。
写入	00	FRCK2_0 用作自由运行计时器 FRCK2 的输入引脚。[初始值]
	01	与写入 00 时相同。
	10	FRCK2_1 用作自由运行计时器 FRCK2 的输入引脚。
	11	禁止设置。

[bit17:16] DTTI2S：DTTI2X 输入选择位

选择 DTTI2X 输入。

bit17:16		描述
读取		读取寄存器值。
写入	00	DTTI2X_0 用作波形发生器 DTTI2X 的输入引脚。[初始值]
	01	与写入 00 时相同。
	10	DTTI2X_1 用作波形发生器 DTTI2X 的输入引脚。
	11	禁止设置。

[bit15:13] 保留：保留位

这些位的读取值为 "0b000"。

写入时，将这些位设置为 "0b000"。

[bit12] DTTI2C： DTTI2X 功能选择位

选择 DTTI2X 功能。

位		描述
读取		读取寄存器值。
写入	0	从 RTO20 至 RTO25 引脚输出，不用 DTTIF2 转换 GPIO。[初始值]
	1	通过 DTTIF2 转换 GPIO 进行引脚 RTO20 至 RTO25 输出。

[bit11:10] RTO25E：RTO25 输出选择位

选择 RTO25 输出。

bit11:10		描述
读取		读取寄存器值。
写入	00	不产生波形发生器 RTO25 的输出。[初始值]
	01	RTO25_0 用作波形发生器 RTO25 的输出引脚。
	10	RTO25_1 用作波形发生器 RTO25 的输出引脚。
	11	禁止设置。

[bit9:8] RTO24E：RTO24 输出选择位

选择 RTO24 输出。

bit9:8		描述
读取		读取寄存器值。
写入	00	不产生波形发生器 RTO24 的输出。[初始值]
	01	RTO24_0 用作波形发生器 RTO24 的输出引脚。
	10	RTO24_1 用作波形发生器 RTO24 的输出引脚。
	11	禁止设置。

[bit7:6] RTO23E：RTO23 输出选择位

选择 RTO23 输出。

bit7:6		描述
读取		读取寄存器值。
写入	00	不产生输出波形发生器 RTO23 的输出。[初始值]
	01	RTO23_0 用作波形发生器 RTO23 的输出引脚。
	10	RTO23_1 用作波形发生器 RTO23 的输出引脚。
	11	禁止设置。

[bit5:4] RTO22E：RTO22 输出选择位

选择 RTO22 输出。

bit5:4		描述
读取		读取寄存器值。
写入	00	不产生波形发生器 RTO22 的输出。[初始值]
	01	RTO22_0 用作波形发生器 RTO22 的输出引脚。
	10	RTO22_1 用作波形发生器 RTO22 的输出引脚。
	11	禁止设置。

[bit3:2] RTO21E：RTO21 输出选择位

选择 RTO21 输出。

bit3:2		描述
读取		读取寄存器值。
写入	00	不产生波形发生器 RTO21 的输出。[初始值]
	01	RTO21_0 用作波形发生器 RTO21 的输出引脚。
	10	RTO21_1 用作波形发生器 RTO21 的输出引脚。
	11	禁止设置。

[bit1:0] RTO20E：RTO20 输出选择位

选择 RTO20 输出。

bit1:0		描述
读取		读取寄存器值。
写入	00	不产生波形发生器 RTO20 的输出。[初始值]
	01	RTO20_0 用作波形发生器 RTO20 的输出引脚。
	10	RTO20_1 用作波形发生器 RTO20 的输出引脚。
	11	禁止设置。

4.12 扩展引脚功能设置寄存器 04 (EPFR04)

EPFR04 寄存器分配基本计时器 ch.0、ch.1、ch.2 和 ch.3 引脚的功能。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	保留		TIOB3S		TIOA3E		TIOA3S	
属性	-		R/W		R/W		R/W	
初始值	-		00		00		00	

位	23	22	21	20	19	18	17	16
字段	保留		TIOB2S		TIOA2E		保留	
属性	-		R/W		R/W		-	
初始值	-		00		00		-	

位	15	14	13	12	11	10	9	8
字段	保留		TIOB1S		TIOA1E		TIOA1S	
属性	-		R/W		R/W		R/W	
初始值	-		00		00		00	

位	7	6	5	4	3	2	1	0
字段	保留	TIOB0S			TIOA0E		保留	
属性	-	R/W			R/W		-	
初始值	-	000			00		-	

寄存器功能

[bit31:30] 保留：保留位

这些位的读取值为 "0b00"。

写入这些位时，将其设置为 "0b00"。

[bit29:28] TIOB3S： TIOB3 输入选择位

选择 TIOB3 输入。

bit29:28		描述
读取		读取寄存器值。
写入	00	BTch.3 TIOB 输入引脚使用 TIOB3_0。[初始值]
	01	与写入 00 时相同。
	10	BTch.3 TIOB 输入引脚使用 TIOB3_1。
	11	BTch.3 TIOB 输入引脚使用 TIOB3_2。

[bit27:26] TIOA3E：TIOA3 输出选择位

选择 TIOA3 输出。

bit27:26		描述
读取		读取寄存器值。
写入	00	不产生 BTch.3 TIOA 的输出。[初始值]
	01	BTch.3 TIOA 输出引脚使用 TIOA3_0。
	10	BTch.3 TIOA 输出引脚使用 TIOA3_1。
	11	BTch.3 TIOA 输出引脚使用 TIOA3_2。

[bit25:24] TIOA3S：TIOA3 输入选择位

选择 TIOA3 输入。

bit25:24		描述
读取		读取寄存器值。
写入	00	BTch.3 TIOA 输入引脚使用 TIOA3_0。[初始值]
	01	与写入 00 时相同。
	10	BTch.3 TIOA 输入引脚使用 TIOA3_1。
	11	BTch.3 TIOA 输入引脚使用 TIOA3_2。

[bit23:22] 保留：保留位

这些位的读取值为 "0b00"。

写入这些位时，将其设置为 "0b00"。

[bit21:20] TIOB2S：TIOB2 输入选择位

选择 TIOB2 输入。

bit21:20		描述
读取		读取寄存器值。
写入	00	BTch.2 TIOB 输入引脚使用 TIOB2_0。[初始值]
	01	与写入 00 时相同。
	10	BTch.2 TIOB 输入引脚使用 TIOB2_1。
	11	BTch.2 TIOB 输入引脚使用 TIOB2_2。

[bit19:18] TIOA2E：TIOA2 输出选择位

选择 TIOA2 输出。

bit19:18		描述
读取		读取寄存器值。
写入	00	不产生 BTch.2 TIOA 的输出。[初始值]
	01	BTch.2 TIOA 输出引脚使用 TIOA2_0。
	10	BTch.2 TIOA 输出引脚使用 TIOA2_1。
	11	BTch.2 TIOA 输出引脚使用 TIOA2_2。

[bit17:14] 保留：保留位

这些位的读取值为 "0b0000"。

写入这些位时，将其设置为 "0b0000"。

[bit13:12] TIOB1S：TIOB1 输入选择位

选择 TIOB1 输入。

bit13:12		描述
读取		读取寄存器值。
写入	00	BTch.1 TIOB 输入引脚使用 TIOB1_0。[初始值]
	01	与写入 00 时相同。
	10	BTch.1 TIOB 输入引脚使用 TIOB1_1。
	11	BTch.1 TIOB 输入引脚使用 TIOB1_2。

[bit11:10] TIOA1E：TIOA1 输出选择位

选择 TIOA1 输出。

bit11:10		描述
读取		读取寄存器值。
写入	00	不产生 BTch.1 TIOA 的输出。 [初始值]
	01	BTch.1 TIOA 输出引脚使用 TIOA1_0。
	10	BTch.1 TIOA 输出引脚使用 TIOA1_1。
	11	BTch.1 TIOA 输出引脚使用 TIOA1_2。

[bit9:8] TIOA1S：TIOA1 输入选择位

选择 TIOA1 输入。

bit9:8		描述
读取		读取寄存器值。
写入	00	BTch.1 TIOA 输入引脚使用 TIOA1_0。[初始值]
	01	与写入 00 时相同。
	10	BTch.1 TIOA 输入引脚使用 TIOA1_1。
	11	BTch.1 TIOA 输入引脚使用 TIOA1_2。

[bit7] 保留：保留位

此位读取值为 "0b0"。

写入此位时，将其设置为 "0b0"。

[bit6:4] TIOB0S：TIOB0 输入选择位

选择 TIOB0 输入。

bit6:4		描述
读取		读取寄存器值。
写入	000	BTch.0 TIOB 输入引脚使用 TIOB0_0。[初始值]
	001	与写入 000 时相同。
	010	BTch.0 TIOB 输入引脚使用 TIOB0_1。
	011	BTch.0 TIOB 输入引脚使用 TIOB0_2。
	100	禁止设置。
	101	禁止设置。
	110	BTch.0 TIOB 输入引脚使用 SUBOUT。
	111	在引脚使用，用于测量高速 CR 分频时钟调节。

[bit3:2] TIOA0E：TIOA0 输出选择位

选择 TIOA0 输出。

bit3:2		描述
读取		读取寄存器值。
写入	00	产生 BTch.0 TIOA 的输出。[初始值]
	01	BTch.0 TIOA 输出引脚使用 TIOA0_0。
	10	BTch.0 TIOA 输出引脚使用 TIOA0_1。
	11	BTch.0 TIOA 输出引脚使用 TIOA0_2。

[bit1:0] 保留：保留位

这些位的读取值为 "0b00"。

写入这些位时，将其设置为 "0b00"。

注意事项：

- *TIOA*

  偶数通道仅用于输出。

  奇数通道可用于输入和输出。

- *TIOB*

  仅输入。

- *TIOA1、TIOA3、TIOA5、TIOA7（"A"的奇数编号）不是双向引脚，因此，要么用作输入引脚，要么用作输出引脚。*

  TIOA 奇数通道被选为输出通道时，输入设置将被忽略。

  示例 1： TIOA1 用作输出引脚时：

  当 TIOA1 输出至 TIOA1_0 时，选择 EPFR04:TIOA1E = 01。

  当 TIOA1 输出至 TIOA1_1 时，选择 EPFR04:TIOA1E = 10。

  当 TIOA1 输出至 TIOA1_2 时，选择 EPFR04:TIOA1E = 11。

  则 EPFR04:TIOA1S 设置将被忽略。

  所选引脚设置 ADE=0、PFR=1（DDR 将被忽略）。

  所选引脚同时也使用的其他外设功能引脚的所有输出必须关闭。

  示例 2： TIOA1 用作输入引脚时：

  选择 EPFR04:TIOA1E = 00。

  当 TIOA1 接收 TIOA1_0 输入时，选择 EPFR04:TIOA1S = 00 或 01。

  当 TIOA1 接收 TIOA1_1 输入时，选择 EPFR04:TIOA1S = 10。

  当 TIOA1 接收 TIOA1_2 输入时，选择 EPFR04:TIOA1S = 11。

  所选引脚设置 ADE=0, PFR=1（DDR 将被忽略）。

  所选引脚同时也使用的其他外设功能引脚的所有输出必须关闭。

  ** 当引脚被设置为输入时，所选引脚（GPIO、其他外设功能引脚）同时也使用的引脚输出可作为反馈（通过上文未说明的设置）输入至基本计时器。*

4.13 扩展引脚功能设置寄存器 05 (EPFR05)

EPFR05 寄存器分配基本计时器 ch.4、ch.5、ch.6 和 ch.7 引脚的功能。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	保留		TIOB7S		TIOA7E		TIOA7S	
属性	-		R/W		R/W		R/W	
初始值	-		00		00		00	

位	23	22	21	20	19	18	17	16
字段	保留		TIOB6S		TIOA6E		保留	
属性	-		R/W		R/W		-	
初始值	-		00		00		-	

位	15	14	13	12	11	10	9	8
字段	保留		TIOB5S		TIOA5E		TIOA5S	
属性	-		R/W		R/W		R/W	
初始值	-		00		00		00	

位	7	6	5	4	3	2	1	0
字段	保留		TIOB4S		TIOA4E		保留	
属性	-		R/W		R/W		-	
初始值	-		00		00		-	

寄存器功能

[bit31:30] 保留：保留位

这些位的读取值为 "0b00"。

写入这些位时，将其设置为 "0b00"。

[bit29:28] TIOB7S：TIOB7 输入选择位

选择 TIOB7 输入。

bit29:28		描述
读取		读取寄存器值。
写入	00	BTch.7TIOB 输入引脚使用 TIOB7_0。[初始值]
	01	与写入 00 时相同。
	10	BTch.7TIOB 输入引脚使用 TIOB7_1。
	11	BTch.7TIOB 输入引脚使用 TIOB7_2。

[bit27:26] TIOA7E：TIOA7 输出选择位

选择 TIOA7 输出。

bit27:26		描述
读取		读取寄存器值。
写入	00	不产生 BTch.7TIOA 的输出。[初始值]
	01	BT ch.7 TIOA 输出引脚使用 TIOA7_0。
	10	BT ch.7 TIOA 输出引脚使用 TIOA7_1。
	11	BT ch.7 TIOA 输出引脚使用 TIOA7_2。

[bit25:24] TIOA7S：TIOA7 输入选择位

选择 TIOA7 输入。

bit25:24		描述
读取		读取寄存器值。
写入	00	BT ch.7 TIOA 输入引脚使用 TIOA7_0。[初始值]
	01	与写入 00 时相同。
	10	BT ch.7 TIOA 输入引脚使用 TIOA7_1。
	11	BT ch.7 TIOA 输入引脚使用 TIOA7_2。

[bit23:22] 保留：保留位

这些位的读取值为 "0b00"。

写入这些位时，将其设置为 "0b00"。

[bit21:20] TIOB6S：TIOB6 输入选择位

选择 TIOB6 输入。

bit21:20		描述
读取		读取寄存器值。
写入	00	BT ch.6 TIOB 的输入引脚使用 TIOB6_0。[初始值]
	01	与写入 00 时相同。
	10	BT ch.6 TIOB 的输入引脚使用 TIOB6_1。
	11	BT ch.6 TIOB 的输入引脚使用 TIOB6_2。

[bit19:18] TIOA6E：TIOA6 输出选择位

选择 TIOA6 输出。

bit19:18		描述
读取		读取寄存器值。
写入	00	不产生 BTch.6TIOA 的输出。[初始值]
	01	BT ch.6 TIOA 输出引脚使用 TIOA6_0。
	10	BT ch.6 TIOA 输出引脚使用 TIOA6_1。
	11	BT ch.6 TIOA 输出引脚使用 TIOA6_2。

[bit17:14] 保留：保留位

这些位的读取值为 "0b0000"。

写入这些位时，将其设置为 "0b0000"。

[bit13:12] TIOB5S：TIOB5 输入选择位

选择 TIOB5 输入。

bit13:12		描述
读取		读取寄存器值。
写入	00	BTch.5 TIOB 的输入引脚使用 TIOB5_0。[初始值]
	01	与写入 00 时相同。
	10	BTch.5 TIOB 的输入引脚使用 TIOB5_1。
	11	BTch.5 TIOB 的输入引脚使用 TIOB5_2。

[bit11:10] TIOA5E：TIOA5 输出选择位

选择 TIOA5 输出。

bit11:10		描述
读取		读取寄存器值。
写入	00	不产生 BTch.5TIOA 的输出。[初始值]
	01	BTch.5 TIOA 的输出引脚使用 TIOA5_0。
	10	BTch.5 TIOA 的输出引脚使用 TIOA5_1。
	11	BTch.5 TIOA 的输出引脚使用 TIOA5_2。

[bit9:8] TIOA5S：TIOA5 输入选择位

选择 TIOA5 输入。

bit9:8		描述
读取		读取寄存器值。
写入	00	BTch.5 TIOA 的输入引脚使用 TIOA5_0。[初始值]
	01	与写入 00 时相同。
	10	BTch.5 TIOA 的输入引脚使用 TIOA5_1。
	11	BTch.5 TIOA 的输入引脚使用 TIOA5_2。

[bit7:6] 保留：保留位

这些位的读取值为 "0b00"。

写入这些位时，将其设置为 "0b00"。

[bit5:4] TIOB4S：TIOB4 输入选择位

选择 TIOB4 输入。

bit5:4		描述
读取		读取寄存器值。
写入	00	BTch.4 TIOB 的输入引脚使用 TIOB4_0。[初始值]
	01	与写入 00 时相同。
	10	BTch.4 TIOB 的输入引脚使用 TIOB4_1。
	11	BTch.4 TIOB 的输入引脚使用 TIOB4_2。

[bit3:2] TIOA4E：TIOA4 输出选择位

选择 TIOA4 输出。

bit3:2		描述
读取		读取寄存器值。
写入	00	不产生 BTch.4 TIOA 的输出。[初始值]
	01	BTch.4 TIOA 输出引脚使用 TIOA4_0。
	10	BTch.4 TIOA 输出引脚使用 TIOA4_1。
	11	BTch.4 TIOA 输出引脚使用 TIOA4_2。

[bit1:0] 保留：保留位

这些位的读取值为 "0b00"。
写入这些位时，将其设置为 "0b00"。

注意事项：

- *TIOA*
  偶数通道仅用于输出。
  奇数通道也可用于输入和输出。
- *TIOB*
  仅输入。
- *TIOA1、TIOA3、TIOA5、TIOA7（"A"的奇数编号）不是双向引脚，因此，要么用作输入引脚，要么用作输出引脚。*
  TIOA 奇数通道被选为输出通道时，输入设置将被忽略。

  示例 1： TIOA1 用作输出引脚时：
  当 TIOA1 输出至 TIOA1_0 时，选择 EPFR04:TIOA1E = 01。
  当 TIOA1 输出至 TIOA1_1 时，选择 EPFR04:TIOA1E = 10。
  当 TIOA1 输出至 TIOA1_2 时，选择 EPFR04:TIOA1E = 11。
  则 EPFR04:TIOA1S 设置将被忽略。
  所选引脚设置 ADE=0, PFR=1（DDR 将被忽略）。
  所选引脚同时也使用的其他外设功能引脚的所有输出必须关闭。

  示例 2：TIOA1 用作输入引脚时：
  选择 EPFR04:TIOA1E = 00。
  当 TIOA1 接收 TIOA1_0 输入时，选择 EPFR04:TIOA1S = 00 或 01。
  当 TIOA1 接收 TIOA1_1 输入时，选择 EPFR04:TIOA1S = 10。
  当 TIOA1 接收 TIOA1_2 输入时，选择 EPFR04:TIOA1S = 11。
  所选引脚设置 ADE=0, PFR=1（DDR 将被忽略）。
  所选引脚同时也使用的其他外设功能引脚的所有输出必须关闭。

  ** 当引脚被设置为输入时，所选引脚（GPIO、其他外设功能引脚）同时也使用的引脚输出可作为反馈（通过上文未说明的设置）输入至基本计时器。*

4.14 扩展引脚功能设置寄存器 06 (EPFR06)

EPFR06 寄存器分配外部中断引脚的功能。

寄存器配置

位	31 30	29 28	27 26	25 24
字段	EINT15S	EINT14S	EINT13S	EINT12S
属性	R/W	R/W	R/W	R/W
初始值	00	00	00	00

位	23 22	21 20	19 18	17 16
字段	EINT11S	EINT10S	EINT09S	EINT08S
属性	R/W	R/W	R/W	R/W
初始值	00	00	00	00

位	15 14	13 12	11 10	9 8
字段	EINT07S	EINT06S	EINT05S	EINT04S
属性	R/W	R/W	R/W	R/W
初始值	00	00	00	00

位	7 6	5 4	3 2	1 0
字段	EINT03S	EINT02S	EINT01S	EINT00S
属性	R/W	R/W	R/W	R/W
初始值	00	00	00	00

寄存器功能

[bit31:30] EINT15S：外部中断输入选择位

选择 EINT15 输入。

bit31:30		描述
读取		读取寄存器值。
写入	00	EINTch.15 输入引脚使用 INT15_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.15 输入引脚使用 INT15_1。
	11	EINTch.15 输入引脚使用 INT15_2。

[bit29:28] EINT14S：外部中断输入选择位

选择 EINT14 输入。

bit29:28		描述
读取		读取寄存器值。
写入	00	EINTch.14 输入引脚使用 INT14_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.14 输入引脚使用 INT14_1。
	11	EINTch.14 输入引脚使用 INT14_2。

[bit27:26] EINT13S：外部中断输入选择位

选择 EINT13 输入。

bit27:26		描述
读取		读取寄存器值。
写入	00	EINTch.13 输入引脚使用 INT13_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.13 输入引脚使用 INT13_1。
	11	EINTch.13 输入引脚使用 INT13_2。

[bit25:24] EINT12S：外部中断输入选择位

选择 EINT12 输入。

bit25:24		描述
读取		读取寄存器值。
写入	00	EINTch.12 输入引脚使用 INT12_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.12 输入引脚使用 INT12_1。
	11	EINTch.12 输入引脚使用 INT12_2。

[bit23:22] EINT11S：外部中断输入选择位

选择 EINT11 输入。

bit23:22		描述
读取		读取寄存器值。
写入	00	EINTch.11 输入引脚使用 INT11_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.11 输入引脚使用 INT11_1。
	11	EINTch.11 输入引脚使用 INT11_2。

[bit21:20] EINT10S：外部中断输入选择位

选择 EINT10 输入。

bit21:20		描述
读取		读取寄存器值。
写入	00	EINTch.10 输入引脚使用 INT10_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.10 输入引脚使用 INT10_1。
	11	EINTch.10 输入引脚使用 INT10_2。

[bit19:18] EINT09S：外部中断输入选择位

选择 EINT09 输入。

bit19:18		描述
读取		读取寄存器值。
写入	00	EINTch.9 输入引脚使用 INT09_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.9 输入引脚使用 INT09_1。
	11	EINTch.9 输入引脚使用 INT09_2。

[bit17:16] EINT08S：外部中断输入选择位

选择 EINT08 输入。

bit17:16		描述
读取		读取寄存器值。
写入	00	EINTch.8 输入引脚使用 INT08_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.8 输入引脚使用 INT08_1。
	11	EINTch.8 输入引脚使用 INT08_2。

[bit15:14] EINT07S：外部中断输入选择位

选择 EINT07 输入。

bit15:14		描述
读取		读取寄存器值。
写入	00	EINTch.7 输入引脚使用 INT07_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.7 输入引脚使用 INT07_1。
	11	EINTch.7 输入引脚使用 INT07_2。

[bit13:12] EINT06S：外部中断输入选择位

选择 EINT06 输入。

bit13:12		描述
读取		读取寄存器值。
写入	00	EINTch.6 输入引脚使用 INT06_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.6 输入引脚使用 INT06_1。
	11	EINTch.6 输入引脚使用 INT06_2。

[bit11:10] EINT05S：外部中断输入选择位

选择 EINT05 输入。

bit11:10		描述
读取		读取寄存器值。
写入	00	EINTch.5 输入引脚使用 INT05_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.5 输入引脚使用 INT05_1。
	11	EINTch.5 输入引脚使用 INT05_2。

[bit9:8] EINT04S：外部中断输入选择位

选择 EINT04 输入。

bit9:8		描述
读取		读取寄存器值。
写入	00	EINTch.4 输入引脚使用 INT04_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.4 输入引脚使用 INT04_1。
	11	EINTch.4 输入引脚使用 INT04_2。

[bit7:6] EINT03S：外部中断输入选择位

选择 EINT03 输入。

bit7:6		描述
读取		读取寄存器值。
写入	00	EINTch.3 输入引脚使用 INT03_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.3 输入引脚使用 INT03_1。
	11	EINTch.3 输入引脚使用 INT03_2。

[bit5:4] EINT02S：外部中断输入选择位

选择 EINT02 输入。

bit5:4		描述
读取		读取寄存器值。
写入	00	EINTch.2 输入引脚使用 INT02_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.2 输入引脚使用 INT02_1。
	11	EINTch.2 输入引脚使用 INT02_2。

[bit3:2] EINT01S：外部中断输入选择位

选择 EINT01 输入。

bit3:2		描述
读取		读取寄存器值。
写入	00	EINTch.1 输入引脚使用 INT01_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.1 输入引脚使用 INT01_1。
	11	EINTch.1 输入引脚使用 INT01_2。

[bit1:0] EINT00S：外部中断输入选择位

选择 EINT00 输入。

bit1:0		描述
读取		读取寄存器值。
写入	00	EINTch.0 输入引脚使用 INT00_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.0 输入引脚使用 INT00_1。
	11	EINTch.0 输入引脚使用 INT00_2。

4.15 扩展引脚功能设置寄存器 07 (EPFR07)

EPFR07 寄存器分配多功能串口 ch.0 至 ch.3 的功能。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	保留				SCK3B		SOT3B	
属性	-				R/W		R/W	
初始值	-				00		00	

位	23	22	21	20	19	18	17	16
字段	SIN3S		SCK2B		SOT2B		SIN2S	
属性	R/W		R/W		R/W		R/W	
初始值	00		00		00		00	

位	15	14	13	12	11	10	9	8
字段	SCK1B		SOT1B		SIN1S		SCK0B	
属性	R/W		R/W		R/W		R/W	
初始值	00		00		00		00	

位	7	6	5	4	3	2	1	0
字段	SOT0B		SIN0S		保留			
属性	R/W		R/W		-			
初始值	00		00		-			

寄存器功能

[bit31:28] 保留：保留位

这些位的读取值为 "0b0000"。

写入这些位时，将其设置为 "0b0000"。

[bit27:26] SCK3B： SCK3 输入/输出选择位

选择 SCK3 输入/输出。

bit27:26		描述
读取		读取寄存器值。
写入	00	MFSch.3SCK 输入引脚使用 SCK3_0。 不产生输出。[初始值]
	01	MFSch.3SCK 输入引脚使用 SCK3_0。 输出引脚使用 SCK3_0。
	10	MFSch.3SCK 输入引脚使用 SCK3_1。 输出引脚使用 SCK3_1。
	11	MFSch.3SCK 输入引脚使用 SCK3_2。 输出引脚使用 SCK3_2。

[bit25:24] SOT3B： SOT3 输入/输出选择位

选择 SOT3 输入/输出。

bit25:24		描述
读取		读取寄存器值。
写入	00	MFSch.3SOT 输入引脚使用 SOT3_0。 不产生输出。[初始值]
	01	MFSch.3SOT 输入引脚使用 SOT3_0。 输出引脚使用 SOT3_0。
	10	MFSch.3SOT 输入引脚使用 SOT3_1。 输出引脚使用 SOT3_1。
	11	MFSch.3SOT 输入引脚使用 SOT3_2。 输出引脚使用 SOT3_2。

[bit23:22] SIN3S：SIN3 输入选择位

选择 SIN3 输入。

bit23:22		描述
读取		读取寄存器值。
写入	00	MFSch.3SIN 输入引脚使用 SIN3_0。[初始值]
	01	与写入 00 时相同。
	10	MFSch.3SIN 输入引脚使用 SIN3_1。
	11	MFSch.3SIN 输入引脚使用 SIN3_2。

[bit21:20] SCK2B：SCK2 输入/输出选择位

选择 SCK2 输入/输出。

bit21:20		描述
读取		读取寄存器值。
写入	00	MFSch.2SCK 输入引脚使用 SCK2_0。 不产生输出。[初始值]
	01	MFSch.2SCK 输入引脚使用 SCK2_0。 输出引脚使用 SCK2_0。
	10	MFSch.2SCK 输入引脚使用 SCK2_1。 输出引脚使用 SCK2_1。
	11	MFSch.2SCK 输入引脚使用 SCK2_2。 输出引脚使用 SCK2_2。

[bit19:18] SOT2B：SOT2 输入/输出选择位

选择 SOT2 输入/输出。

bit19:18		描述
读取		读取寄存器值。
写入	00	MFSch.2SOT 输入引脚使用 SOT2_0。 不产生输出。[初始值]
	01	MFSch.2SOT 输入引脚使用 SOT2_0。 输出引脚使用 SOT2_0。
	10	MFSch.2SOT 输入引脚使用 SOT2_1。 输出引脚使用 SOT2_1。
	11	MFSch.2SOT 输入引脚使用 SOT2_2。 输出引脚使用 SOT2_2。

[bit17:16] SIN2S：SIN2 输入选择位

选择 SIN2 输入。

bit17:16		描述
读取		读取寄存器值。
写入	00	MFSch.2SIN 输入引脚使用 SIN2_0。[初始值]
	01	与写入 00 时相同。
	10	MFSch.2SIN 输入引脚使用 SIN2_1。
	11	MFSch.2SIN 输入引脚使用 SIN2_2。

[bit15:14] SCK1B：SCK1 输入/输出选择位

选择 SCK1 输入/输出。

bit15:14		描述
读取		读取寄存器值。
写入	00	MFSch.1SCK 输入引脚使用 SCK1_0。 不产生输出。[初始值]
	01	MFSch.1SCK 输入引脚使用 SCK1_0。 输出引脚使用 SCK1_0。
	10	MFSch.1SCK 输入引脚使用 SCK1_1。 输出引脚使用 SCK1_1。
	11	MFSch.1SCK 输入引脚使用 SCK1_2。 输出引脚使用 SCK1_2。

[bit13:12] SOT1B：SOT1 输入/输出选择位

选择 SOT1 输入/输出。

bit13:12		描述
读取		读取寄存器值。
写入	00	MFSch.1SOT 输入引脚使用 SOT1_0。 不产生输出。[初始值]
	01	MFSch.1SOT 输入引脚使用 SOT1_0。 输出引脚使用 SOT1_0。
	10	MFSch.1SOT 输入引脚使用 SOT1_1。 输出引脚使用 SOT1_1。
	11	MFSch.1SOT 输入引脚使用 SOT1_2。 输出引脚使用 SOT1_2。

[bit11:10] SIN1S：SIN1 输入选择位

选择 SIN1 输入。

bit11:10		描述
读取		读取寄存器值。
写入	00	MFSch.1SIN 输入引脚使用 SIN1_0。[初始值]
	01	与写入 00 时相同。
	10	MFSch.1SIN 输入引脚使用 SIN1_1。
	11	MFSch.1SIN 输入引脚使用 SIN1_2。

[bit9:8] SCK0B：SCK0 输入/输出选择位

选择 SCK0 输入/输出。

bit9:8		描述
读取		读取寄存器值。
写入	00	MFSch.0SCK 输入引脚使用 SCK0_0。 不产生输出。[初始值]
	01	MFSch.0SCK 输入引脚使用 SCK0_0。 输出引脚使用 SCK0_0。
	10	MFSch.0SCK 输入引脚使用 SCK0_1。 输出引脚使用 SCK0_1。
	11	MFSch.0SCK 输入引脚使用 SCK0_2。 输出引脚使用 SCK0_2。

[bit7:6] SOT0B：SOT0 输入/输出选择位

选择 SOT0 输入/输出。描述

bit7:6		描述
读取		读取寄存器值。
写入	00	MFSch.0SOT 输入引脚使用 SOT0_0。 不产生输出。[初始值]
	01	MFSch.0SOT 输入引脚使用 SOT0_0。 输出引脚使用 SOT0_0。
	10	MFSch.0SOT 输入引脚使用 SOT0_1。 输出引脚使用 SOT0_1。
	11	MFSch.0SOT 输入引脚使用 SOT0_2。 输出引脚使用 SOT0_2。

[bit5:4] SIN0S：SIN0 输入选择位

选择 SIN0 输入。

bit5:4		描述
读取		读取寄存器值。
写入	00	MFSch.0SIN 输入引脚使用 SIN0_0。[初始值]
	01	与写入 00 时相同。
	10	MFSch.0SIN 输入引脚使用 SIN0_1。
	11	MFSch.0SIN 输入引脚使用 SIN0_2。

[bit3:0] 保留：保留位

这些位的读取值为 "0b0000"。

写入这些位时，将其设置为 "0b0000"。

4.16 扩展引脚功能设置寄存器 08 (EPFR08)

EPFR08 寄存器分配多功能串口 ch.4 至 ch.7 的功能。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	保留				SCK7B		SOT7B	
属性	-				R/W		R/W	
初始值	-				00		00	

位	23	22	21	20	19	18	17	16
字段	SIN7S		SCK6B		SOT6B		SIN6S	
属性	R/W		R/W		R/W		R/W	
初始值	00		00		00		00	

位	15	14	13	12	11	10	9	8
字段	SCK5B		SOT5B		SIN5S		SCK4B	
属性	R/W		R/W		R/W		R/W	
初始值	00		00		00		00	

位	7	6	5	4	3	2	1	0
字段	SOT4B		SIN4S		CTS4S		RTS4E	
属性	R/W		R/W		R/W		R/W	
初始值	00		00		00		00	

寄存器功能

[bit31:28] 保留：保留位

这些位的读取值为 "0b0000"。

写入这些位时，将其设置为 "0b0000"。

[bit27:26] SCK7B：SCK7 输入/输出选择位

选择 SCK7 输入/输出。

bit27:26		描述
读取		读取寄存器值。
写入	00	MFSch.7SCK 输入引脚使用 SCK7_0。 不产生输出。[初始值]
	01	MFSch.7SCK 输入引脚使用 SCK7_0。 输出引脚使用 SCK7_0。
	10	MFSch.7SCK 输入引脚使用 SCK7_1。 输出引脚使用 SCK7_1。
	11	MFSch.7SCK 输入引脚使用 SCK7_2。 输出引脚使用 SCK7_2。

[bit25:24] SOT7B：SOT7 输入/输出选择位

选择 SOT7 输入/输出。

bit25:24		描述
读取		读取寄存器值。
写入	00	MFSch.7SOT 输入引脚使用 SOT7_0。 不产生输出。[初始值]
	01	MFSch.7SOT 输入引脚使用 SOT7_0。 输出引脚使用 SOT7_0。
	10	MFSch.7SOT 输入引脚使用 SOT7_1。 输出引脚使用 SOT7_1。
	11	MFSch.7SOT 输入引脚使用 SOT7_2。 输出引脚使用 SOT7_2。

[bit23:22] SIN7S：SIN7 输入选择位

选择 SIN7 输入。

bit23:22		描述
读取		读取寄存器值。
写入	00	MFSch.7SIN 输入引脚使用 SIN7_0。[初始值]
	01	与写入 00 时相同。
	10	MFSch.7SIN 输入引脚使用 SIN7_1。
	11	MFSch.7SIN 输入引脚使用 SIN7_2。

[bit21:20] SCK6B：SCK6 输入/输出选择位

选择 SCK6 输入/输出。

bit21:20		描述
读取		读取寄存器值。
写入	00	MFSch.6SCK 输入引脚使用 SCK6_0。 不产生输出。[初始值]
	01	MFSch.6SCK 输入引脚使用 SCK6_0。 输出引脚使用 SCK6_0。
	10	MFSch.6SCK 输入引脚使用 SCK6_1。 输出引脚使用 SCK6_1。
	11	MFSch.6SCK 输入引脚使用 SCK6_2。 输出引脚使用 SCK6_2。

[bit19:18] SOT6B：SOT6 输入/输出选择位

选择 SOT6 输入/输出。

bit19:18		描述
读取		读取寄存器值。
写入	00	MFSch.6SOT 输入引脚使用 SOT6_0。 不产生输出。[初始值]
	01	MFSch.6SOT 输入引脚使用 SOT6_0。 输出引脚使用 SOT6_0。
	10	MFSch.6SOT 输入引脚使用 SOT6_1。 输出引脚使用 SOT6_1。
	11	MFSch.6SOT 输入引脚使用 SOT6_2。 输出引脚使用 SOT6_2。

[bit17:16] SIN6S：SIN6 输入选择位

选择 SIN6 输入。

bit17:16		描述
读取		读取寄存器值。
写入	00	MFSch.6SIN 输入引脚使用 SIN6_0。[初始值]
	01	与写入 00 时相同。
	10	MFSch.6SIN 输入引脚使用 SIN6_1。
	11	MFSch.6SIN 输入引脚使用 SIN6_2。

[bit15:14] SCK5B：SCK5 输入/输出选择位

选择 SCK5 输入/输出。

bit15:14		描述
读取		读取寄存器值。
写入	00	MFSch.5SCK 输入引脚使用 SCK5_0。 不产生输出。[初始值]
	01	MFSch.5SCK 输入引脚使用 SCK5_0。 输出引脚使用 SCK5_0。
	10	MFSch.5SCK 输入引脚使用 SCK5_1。 输出引脚使用 SCK5_1。
	11	MFSch.5SCK 输入引脚使用 SCK5_2。 输出引脚使用 SCK5_2。

[bit13:12] SOT5B： SOT5 输入/输出选择位

选择 SOT5 输入/输出。

bit13:12		描述
读取		读取寄存器值。
写入	00	MFSch.5SOT 输入引脚使用 SOT5_0。 不产生输出。[初始值]
	01	MFSch.5SOT 输入引脚使用 SOT5_0。 输出引脚使用 SOT5_0。
	10	MFSch.5SOT 输入引脚使用 SOT5_1。 输出引脚使用 SOT5_1。
	11	MFSch.5SOT 输入引脚使用 SOT5_2。 输出引脚使用 SOT5_2。

[bit11:10] SIN5S：SIN5 输入选择位

选择 SIN5 输入。

bit11:10		描述
读取		读取寄存器值。
写入	00	MFSch.5SIN 输入引脚使用 SIN5_0。[初始值]
	01	与写入 00 时相同。
	10	MFSch.5SIN 输入引脚使用 SIN5_1。
	11	MFSch.5SIN 输入引脚使用 SIN5_2。

[bit9:8] SCK4B：SCK4 输入/输出选择位

选择 SCK4 输入/输出。

bit9:8		描述
读取		读取寄存器值。
写入	00	MFSch.4SCK 输入引脚使用 SCK4_0。 不产生输出。[初始值]
	01	MFSch.4SCK 输入引脚使用 SCK4_0。 输出引脚使用 SCK4_0。
	10	MFSch.4SCK 输入引脚使用 SCK4_1。 输出引脚使用 SCK4_1。
	11	MFSch.4SCK 输入引脚使用 SCK4_2。 输出引脚使用 SCK4_2。

[bit7:6] SOT4B：SOT4 输入/输出选择位

选择 SOT4 输入/输出。

bit7:6		描述
读取		读取寄存器值。
写入	00	MFSch.4SOT 输入引脚使用 SOT4_0。 不产生输出。[初始值]
	01	MFSch.4SOT 输入引脚使用 SOT4_0。 输出引脚使用 SOT4_0。
	10	MFSch.4SOT 输入引脚使用 SOT4_1。 输出引脚使用 SOT4_1。
	11	MFSch.4SOT 输入引脚使用 SOT4_2。 输出引脚使用 SOT4_2。

[bit5:4] SIN4S：SIN4 输入选择位

选择 SIN4 输入。

bit5:4		描述
读取		读取寄存器值。
写入	00	MFSch.4SIN 输入引脚使用 SIN4_0。[初始值]
	01	与写入 00 时相同。
	10	MFSch.4SIN 输入引脚使用 SIN4_1。
	11	MFSch.4SIN 输入引脚使用 SIN4_2。

[bit3:2] CTS4S：CTS4 输入选择位

选择 CTS4 输入。

bit3:2		描述
读取		读取寄存器值。
写入	00	MFSch.4CTS 输入引脚使用 CTS4_0。[初始值]
	01	与写入 00 时相同。
	10	MFSch.4CTS 输入引脚使用 CTS4_1。
	11	MFSch.4CTS 输入引脚使用 CTS4_2。

[bit1:0] RTS4E：RTS4 输出选择位

选择 RTS4 输出。

bit1:0		描述
读取		读取寄存器值。
写入	00	不产生 MFSch.4RTS 的输出。[初始值]
	01	MFSch.4RTS 输出引脚使用 RTS4_0。
	10	MFSch.4RTS 输出引脚使用 RTS4_1。
	11	MFSch.4RTS 输出引脚使用 RTS4_2。

4.17 扩展引脚功能设置寄存器 09 (EPFR09)

EPFR09 寄存器分配 CAN、ADC 触发信号和 QPRC 外设引脚的功能。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	CTX1E		CRX1S		CTX0E		CRX0S	
属性	R/W		R/W		R/W		R/W	
初始值	00		00		00		00	

位	23	22	21	20	19	18	17	16
字段	ADTRG2S				ADTRG1S			
属性	R/W				R/W			
初始值	0000				0000			

位	15	14	13	12	11	10	9	8
字段	ADTRG0S				QZIN1S		QBIN1S	
属性	R/W				R/W		R/W	
初始值	0000				00		00	

位	7	6	5	4	3	2	1	0
字段	QAIN1S		QZIN0S		QBIN0S		QAIN0S	
属性	R/W		R/W		R/W		R/W	
初始值	00		00		00		00	

寄存器功能

[bit31:30] CTX1E： CTX1E 输出选择位

选择 CAN TX1 输出。

bit31:30		描述
读取		读取寄存器值。
写入	00	不产生 CANch.1TX 的输出。[初始值]
	01	将 CANch.1TX 输出引脚设置为 TX1_0。
	10	将 CANch.1TX 输出引脚设置为 TX1_1。
	11	将 CANch.1TX 输出引脚设置为 TX1_2。

[bit29:28] CRX1S：CRX1S 输入选择位

选择 CAN RX1 输入。

bit29:28		描述
读取		读取寄存器值。
写入	00	将 CANch.1RX 输入引脚设置为 RX1_0。[初始值]
	01	与写入 00 时相同。
	10	将 CANch.1RX 输入引脚设置为 RX1_1。
	11	将 CANch.1RX 输入引脚设置为 RX1_2。

[bit27:26] CTX0E：CTX0E 输出选择位

选择 CAN TX0 输出。

bit27:26		描述
读取		读取寄存器值。
写入	00	不产生 CANch.0TX 的输出。[初始值]
	01	将 CANch.0TX 输出引脚设置为 TX0_0。
	10	将 CANch.0TX 输出引脚设置为 TX0_1。
	11	将 CANch.0TX 输出引脚设置为 TX0_2。

[bit25:24] CRX0S： CRX0S 输入选择位

选择 CAN RX0 输入。

bit25:24		描述
读取		读取寄存器值。
写入	00	将 CANch.0RX 输入引脚设置为 RX0_0。[初始值]
	01	与写入 00 时相同。
	10	将 CANch.0RX 输入引脚设置为 RX0_1。
	11	将 CANch.0RX 输入引脚设置为 RX0_2。

[bit23:20] ADTRG2S：ADTRG2 输入选择位

选择 ADTRG2 输入。

bit23:20		描述
读取		读取寄存器值。
写入	0000	ADC 单元 2 启动触发信号的输入引脚使用 ADTG_0。[初始值]
	0001	与写入 0000 时相同。
	0010	ADC 单元 2 启动触发信号的输入引脚使用 ADTG_1。
	0011	ADC 单元 2 启动触发信号的输入引脚使用 ADTG_2。
	0100	ADC 单元 2 启动触发信号的输入引脚使用 ADTG_3。
	0101	ADC 单元 2 启动触发信号的输入引脚使用 ADTG_4。
	0110	ADC 单元 2 启动触发信号的输入引脚使用 ADTG_5。
	0111	ADC 单元 2 启动触发信号的输入引脚使用 ADTG_6。
	1000	ADC 单元 2 启动触发信号的输入引脚使用 ADTG_7。
	1001	ADC 单元 2 启动触发信号的输入引脚使用 ADTG_8。
写入其他数据		禁止设置。

[bit19:16] ADTRG1S：ADTRG1 输入选择位

选择 ADTRG1 输入。

bit19:16		描述
读取		读取寄存器值。
写入	0000	ADC 单元 1 启动触发信号的输入引脚使用 ADTG_0。[初始值]
	0001	与写入 0000 时相同。
	0010	ADC 单元 1 启动触发信号的输入引脚使用 ADTG_1。
	0011	ADC 单元 1 启动触发信号的输入引脚使用 ADTG_2。
	0100	ADC 单元 1 启动触发信号的输入引脚使用 ADTG_3。
	0101	ADC 单元 1 启动触发信号的输入引脚使用 ADTG_4。
	0110	ADC 单元 1 启动触发信号的输入引脚使用 ADTG_5。
	0111	ADC 单元 1 启动触发信号的输入引脚使用 ADTG_6。
	1000	ADC 单元 1 启动触发信号的输入引脚使用 ADTG_7。
	1001	ADC 单元 1 启动触发信号的输入引脚使用 ADTG_8。
写入其他数据		禁止设置。

[bit15:12] ADTRG0S：ADTRG0 输入选择位

选择 ADTRG0 输入。

bit15:12		描述
读取		读取寄存器值。
写入	0000	ADC 单元 0 启动触发信号的输入引脚使用 ADTG_0。[初始值]
	0001	与写入 0000 时相同。
	0010	ADC 单元 0 启动触发信号的输入引脚使用 ADTG_1。
	0011	ADC 单元 0 启动触发信号的输入引脚使用 ADTG_2。
	0100	ADC 单元 0 启动触发信号的输入引脚使用 ADTG_3。
	0101	ADC 单元 0 启动触发信号的输入引脚使用 ADTG_4。
	0110	ADC 单元 0 启动触发信号的输入引脚使用 ADTG_5。
	0111	ADC 单元 0 启动触发信号的输入引脚使用 ADTG_6。
	1000	ADC 单元 0 启动触发信号的输入引脚使用 ADTG_7。
	1001	ADC 单元 0 启动触发信号的输入引脚使用 ADTG_8。
写入其他数据		禁止设置。

[bit11:10] QZIN1S： QZIN1S 输入选择位

选择 QPRC ZIN1 输入。

bit11:10		描述
读取		读取寄存器值。
写入	00	QPRCch.1's ZIN 输入引脚使用 ZIN1_0。[初始值]
	01	与写入 00 时相同。
	10	QPRCch.1's ZIN 输入引脚使用 ZIN1_1。
	11	QPRCch.1's ZIN 输入引脚使用 ZIN1_2。

[bit9:8] QBIN1S：QBIN1S 输入选择位

选择 QPRC BIN1 输入。

bit9:8		描述
读取		读取寄存器值。
写入	00	QPRCch.1's BIN 输入引脚使用 BIN1_0。[初始值]
	01	与写入 00 时相同。
	10	QPRCch.1's BIN 输入引脚使用 BIN1_1。
	11	QPRCch.1's BIN 输入引脚使用 BIN1_2。

[bit7:6] QAIN1S：QAIN1S 输入选择位

选择 QPRC AIN1 输入。

bit7:6		描述
读取		读取寄存器值。
写入	00	QPRCch.1's AIN 输入引脚使用 AIN1_0。[初始值]
	01	与写入 00 时相同。
	10	QPRCch.1's AIN 输入引脚使用 AIN1_1。
	11	QPRCch.1's AIN 输入引脚使用 AIN1_2。

[bit5:4] QZIN0S：QZIN0S 输入选择位

选择 QPRC ZIN0 输入。如果使用 ZIN0_3，设置此位以及扩展引脚功能设置寄存器 21 (EPFR21) 中的 QZIN0S[2] 位。

EPFR21[2],bit5:4		描述
读取		读取寄存器值。
写入	000	QPRCch.0's ZIN 输入引脚使用 ZIN0_0。[初始值]
	001	与写入 00 时相同。
	010	QPRCch.0's ZIN 输入引脚使用 ZIN0_1。
	011	QPRCch.0's ZIN 输入引脚使用 ZIN0_2。
	100	QPRCch.0's ZIN 输入引脚使用 ZIN0_3。
	其它	禁止设置。

[bit3:2] QBIN0S：QBIN0S 输入选择位

选择 QPRC BIN0 输入。如果使用 BIN0_3，则设置此位以及扩展引脚功能设置寄存器 21 (EPFR21) 中的 QBIN0S[2] 位。

EPFR21[1],bit3:2		描述
读取		读取寄存器值。
写入	000	QPRCch.0's BIN 输入引脚使用 BIN0_0。[初始值]
	001	与写入 00 时相同。
	010	QPRCch.0's BIN 输入引脚使用 BIN0_1。
	011	QPRCch.0's BIN 输入引脚使用 BIN0_2。
	100	QPRCch.0's BIN 输入引脚使用 BIN0_3。
	其它	禁止设置

[bit1:0] QAIN0S：QAIN0S 输入选择位

选择 QPRC AIN0 输入。如果使用 AIN0_3，设置此位以及扩展引脚功能设置寄存器 21 (EPFR21) 中的 QAIN0S[2] 位。

EPFR21[0],bit1:0		描述
读取		读取寄存器值。
写入	000	QPRCch.0's AIN 输入引脚使用 AIN0_0。[初始值]
	001	与写入 00 时相同。
	010	QPRCch.0's AIN 输入引脚使用 AIN0_1。
	011	QPRCch.0's AIN 输入引脚使用 AIN0_2。
	100	QPRCch.0's AIN 输入引脚使用 AIN0_3。
	其它	禁止设置。

4.18 扩展引脚功能设置寄存器 12 (EPFR12)

EPFR12 寄存器分配基本计时器 ch.8、ch.9、ch.10 和 ch.11 引脚的功能。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	保留		TIOB11S		TIOA11E		TIOA11S	
属性	-		R/W		R/W		R/W	
初始值	-		00		00		00	

位	23	22	21	20	19	18	17	16
字段	保留		TIOB10S		TIOA10E		保留	
属性	-		R/W		R/W		-	
初始值	-		00		00		-	

位	15	14	13	12	11	10	9	8
字段	保留		TIOB9S		TIOA9E		TIOA9S	
属性	-		R/W		R/W		R/W	
初始值	-		00		00		00	

位	7	6	5	4	3	2	1	0
字段	保留		TIOB8S		TIOA8E		保留	
属性	-		R/W		R/W		-	
初始值	-		00		00		-	

寄存器功能

[bit31:30] 保留：保留位

这些位的读取值为 "0b00"。

写入这些位时，将其设置为 "0b00"。

[bit29:28] TIOB11S： TIOB11 输入选择位

选择 TIOB11 输入。

位		描述
读取		读取寄存器值。
写入	00	BTch.11 TIOB 输入引脚使用 TIOB11_0。[初始值]
	01	与写入 00 时相同。
	10	BTch.11 TIOB 输入引脚使用 TIOB11_1。
	11	BTch.11 TIOB 输入引脚使用 TIOB11_2。

[bit27:26] TIOA11E：TIOA11 输出选择位

选择 TIOA11 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 BTch.11 TIOA 的输出。[初始值]
	01	BT ch.11 TIOA 输出引脚使用 TIOA11_0。
	10	BT ch.11 TIOA 输出引脚使用 TIOA11_1。
	11	BT ch.11 TIOA 输出引脚使用 TIOA11_2。

[bit25:24] TIOA11S： TIOA11 输入选择位

选择 TIOA11 输入。

位		描述
读取		读取寄存器值。
写入	00	BT ch.11 TIOA 输入引脚使用 TIOA11_0。[初始值]
	01	与写入 00 时相同。
	10	BT ch.11 TIOA 输入引脚使用 TIOA11_1。
	11	BT ch.11 TIOA 输入引脚使用 TIOA11_2。

[bit23:22] 保留：保留位

这些位的读取值为 "0b00"。

写入这些位时，将其设置为 "0b00"。

[bit21:20] TIOB10S： TIOB10 输入选择位

选择 TIOB10 输入。

位		描述
读取		读取寄存器值。
写入	00	BT ch.10 TIOB 的输入引脚使用 TIOB10_0。[初始值]
	01	与写入 00 时相同。
	10	BT ch.10 TIOB 的输入引脚使用 TIOB10_1。
	11	BT ch.10 TIOB 的输入引脚使用 TIOB10_2。

[bit19:18] TIOA10E：TIOA10 输出选择位

选择 TIOA10 输出。

bit19:18		描述
读取		读取寄存器值。
写入	00	不产生 BTch.10 TIOA 的输出。[初始值]
	01	BT ch.10 TIOA 输出引脚使用 TIOA10_0。
	10	BT ch.10 TIOA 输出引脚使用 TIOA10_1。
	11	BT ch.10 TIOA 输出引脚使用 TIOA10_2。

[bit17:14] 保留：保留位

这些位的读取值为 "0b0000"。

写入这些位时，将其设置为 "0b0000"。

[bit13:12] TIOB9S： TIOB9 输入选择位

选择 TIOB9 输入。

位		描述
读取		读取寄存器值。
写入	00	BTch.9 TIOB 输入引脚使用 TIOB9_0。[初始值]
	01	与写入 00 时相同。
	10	BTch.9 TIOB 输入引脚使用 TIOB9_1。
	11	BTch.9 TIOB 输入引脚使用 TIOB9_2。

[bit11:10] TIOA9E：TIOA9 输出选择位

选择 TIOA9 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 BTch.9 TIOA 的输出。[初始值]
	01	BTch.9 TIOA 输出引脚使用 TIOA9_0。
	10	BTch.9 TIOA 输出引脚使用 TIOA9_1。
	11	BTch.9 TIOA 输出引脚使用 TIOA9_2。

[bit9:8] TIOA9S：TIOA9 输入选择位

选择 TIOA9 输入。

位		描述
读取		读取寄存器值。
写入	00	BTch.9 TIOA 输入引脚使用 TIOA9_0。[初始值]
	01	与写入 00 时相同。
	10	BTch.9 TIOA 输入引脚使用 TIOA9_1。
	11	BTch.9 TIOA 输入引脚使用 TIOA9_2。

[bit7:6] 保留：保留位

这些位的读取值为 "0b00"。

写入这些位时，将其设置为 "0b00"。

[bit5:4] TIOB8S：TIOB8 输入选择位

选择 TIOB8 输入。

bit5:4		描述
读取		读取寄存器值。
写入	00	BTch.8 TIOB 输入引脚使用 TIOB8_0。[初始值]
	01	与写入 00 时相同。
	10	BTch.8 TIOB 输入引脚使用 TIOB8_1。
	11	BTch.8 TIOB 输入引脚使用 TIOB8_2。

[bit3:2] TIOA8E： TIOA8 Output Select bits

选择 TIOA8 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 BTch.8 TIOA 的输出。[初始值]
	01	BTch.8 TIOA 输出引脚使用 TIOA8_0。
	10	BTch.8 TIOA 输出引脚使用 TIOA8_1。

	11	BTch.8 TIOA 输出引脚使用 TIOA8_2。

[bit1:0] 保留：保留位

这些位的读取值为 "0b00"。
写入这些位时，将其设置为 "0b00"。

注意事项：

- *TIOA*
  偶数通道仅用于输出。
  奇数通道也可用于输入和输出。
- *TIOB*
  仅输入。
- *TIOA9、TIOA11、TIOA13、TIOA15（"A"的奇数编号）不是双向引脚，因此，要么用作输入引脚，要么用作输出引脚。*
  TIOA 奇数通道被选为输出通道时，输入设置将被忽略。

  示例 1： TIOA11 用作输出引脚时：
  当 TIOA11 输出至 TIOA11_0 时，选择 EPFR12:TIOA11E = 01。
  当 TIOA11 输出至 TIOA11_1 时，选择 EPFR12:TIOA11E = 10。
  当 TIOA11 输出至 TIOA11_2 时，选择 EPFR12:TIOA11E = 11。
  EPFR12:TIOA11S 设置将被忽略。
  所选引脚设置 ADE=0, PFR=1（DDR 将被忽略）。
  所选引脚同时也使用的其他外设功能引脚的所有输出必须关闭。

  示例 2：TIOA11 用作输入引脚时：
  选择 EPFR12:TIOA11E = 00。
  当 TIOA11 接收 TIOA11_0 输入时，选择 EPFR12:TIOA11S = 00 或 01。
  当 TIOA11 接收 TIOA11_1 输入时，选择 EPFR12:TIOA11S = 10。
  当 TIOA11 接收 TIOA11_2 输入时，选择 EPFR12:TIOA11S = 11。
  所选引脚设置 ADE=0, PFR=1（DDR 将被忽略）。
  所选引脚同时也使用的其他外设功能引脚的所有输出必须关闭。

  ** 当引脚被设置为输入时，所选引脚（GPIO、其他外设功能引脚）同时也使用的引脚输出可作为反馈（通过上文未说明的设置）输入至基本计时器。*

4.19 扩展引脚功能设置寄存器 13 (EPFR13)

EPFR13 寄存器分配基本计时器 ch.12、ch.13、ch.14 和 ch.15 引脚的功能。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	保留		TIOB15S		TIOA15E		TIOA15S	
属性	-		R/W		R/W		R/W	
初始值	-		00		00		00	

位	23	22	21	20	19	18	17	16
字段	保留		TIOB14S		TIOA14E		保留	
属性	-		R/W		R/W		-	
初始值	-		00		00		-	

位	15	14	13	12	11	10	9	8
字段	保留		TIOB13S		TIOA13E		TIOA13S	
属性	-		R/W		R/W		R/W	
初始值	-		00		00		00	

位	7	6	5	4	3	2	1	0
字段	保留		TIOB12S		TIOA12E		保留	
属性	-		R/W		R/W		-	
初始值	-		00		00		-	

寄存器功能

[bit31:30] 保留：保留位

这些位的读取值为 "0b00"。

写入这些位时，将其设置为 "0b00"。

[bit29:28] TIOB15S：TIOB15 输入选择位

选择 TIOB15 输入。

位		描述
读取		读取寄存器值。
写入	00	BTch.15 TIOB 输入引脚使用 TIOB15_0。[初始值]
	01	与写入 00 时相同。
	10	BTch.15 TIOB 输入引脚使用 TIOB15_1。
	11	BTch.15 TIOB 输入引脚使用 TIOB15_2。

[bit27:26] TIOA15E：TIOA15 输出选择位

选择 TIOA15 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 BTch.15 TIOA 的输出。[初始值]
	01	BT ch.15 TIOA 输出引脚使用 TIOA15_0。
	10	BT ch.15 TIOA 输出引脚使用 TIOA15_1。
	11	BT ch.15 TIOA 输出引脚使用 TIOA15_2。

[bit25:24] TIOA15S：TIOA15 输入选择位

选择 TIOA15 输入。

位		描述
读取		读取寄存器值。
写入	00	BT ch.15 TIOA 输入引脚使用 TIOA15_0。[初始值]
	01	与写入 00 时相同。
	10	BT ch.15 TIOA 输入引脚使用 TIOA15_1。
	11	BT ch.15 TIOA 输入引脚使用 TIOA15_2。

[bit23:22] 保留：保留位

这些位的读取值为 "0b00"。

写入这些位时，将其设置为 "0b00"。

[bit21:20] TIOB14S： TIOB14 输入选择位

选择 TIOB14 输入。

bit21:20		描述
读取		读取寄存器值。
写入	00	BT ch.14 TIOB 的输入引脚使用 TIOB14_0。[初始值]
	01	与写入 00 时相同。
	10	BT ch.14 TIOB 的输入引脚使用 TIOB14_1。
	11	BT ch.14 TIOB 的输入引脚使用 TIOB14_2。

[bit19:18] TIOA14E：TIOA14 输出选择位

选择 TIOA14 输出。

bit19:18		描述
读取		读取寄存器值。
写入	00	不产生 BTch.14 TIOA 的输出。[初始值]
	01	BT ch.14 TIOA 输出引脚使用 TIOA14_0。
	10	BT ch.14 TIOA 输出引脚使用 TIOA14_1。
	11	BT ch.14 TIOA 输出引脚使用 TIOA14_2。

[bit17:14] 保留：保留位

这些位的读取值为 "0b0000"。

写入这些位时，将其设置为 "0b0000"。

[bit13:12] TIOB13S： TIOB13 输入选择位

选择 TIOB13 输入。

bit13:12		描述
读取		读取寄存器值。
写入	00	BTch.13 TIOB 输入引脚使用 TIOB13_0。[初始值]
	01	与写入 00 时相同。
	10	BTch.13 TIOB 输入引脚使用 TIOB13_1。
	11	BTch.13 TIOB 输入引脚使用 TIOB13_2。

[bit11:10] TIOA13E：TIOA13 输出选择位

选择 TIOA13 输出。

bit11:10		描述
读取		读取寄存器值。
写入	00	不产生 BTch.13 TIOA 的输出。[初始值]
	01	BTch.13 TIOA 输出引脚使用 TIOA13_0。
	10	BTch.13 TIOA 输出引脚使用 TIOA13_1。
	11	BTch.13 TIOA 输出引脚使用 TIOA13_2。

[bit9:8] TIOA13S：TIOA13 输入选择位

选择 TIOA13 输入。

bit9:8		描述
读取		读取寄存器值。
写入	00	BTch.13 TIOA 输入引脚使用 TIOA13_0。[初始值]
	01	与写入 00 时相同。
	10	BTch.13 TIOA 输入引脚使用 TIOA13_1。
	11	BTch.13 TIOA 输入引脚使用 TIOA13_2。

[bit7:6] 保留：保留位

这些位的读取值为 "0b00"。

写入这些位时，将其设置为 "0b00"。

[bit5:4] TIOB12S：TIOB12 输入选择位

选择 TIOB12 输入。

bit5:4		描述
读取		读取寄存器值。
写入	00	BTch.12 TIOB 输入引脚使用 TIOB12_0。[初始值]
	01	与写入 00 时相同。
	10	BTch.12 TIOB 输入引脚使用 TIOB12_1。
	11	BTch.12 TIOB 输入引脚使用 TIOB12_2。

[bit3:2] TIOA12E：TIOA12 输出选择位

选择 TIOA12 输出。

bit3:2		描述
读取		读取寄存器值。
写入	00	不产生 BTch.12 TIOA 的输出。[初始值]
	01	BTch.12 TIOA 输出引脚使用 TIOA12_0。

	10	BTch.12 TIOA 输出引脚使用 TIOA12_1。
	11	BTch.12 TIOA 输出引脚使用 TIOA12_2。

[bit1:0] 保留：保留位

这些位的读取值为 "0b00"。
写入这些位时，将其设置为 "0b00"。

注意事项：

- *TIOA*
  偶数通道仅用于输出。
  奇数通道也可用于输入和输出。
- *TIOB*
  仅输入。
- *TIOA9、TIOA11、TIOA13、TIOA15（"A"的奇数编号）不是双向引脚，因此，要么用作输入引脚，要么用作输出引脚。*
  TIOA 奇数通道被选为输出通道时，输入设置将被忽略。

  示例 1： TIOA11 用作输出引脚时：
  当 TIOA11 输出至 TIOA11_0 时，选择 EPFR12:TIOA11E = 01。
  当 TIOA11 输出至 TIOA11_1 时，选择 EPFR12:TIOA11E = 10。
  当 TIOA11 输出至 TIOA11_2 时，选择 EPFR12:TIOA11E = 11。
  EPFR12:TIOA11S 设置将被忽略。
  所选引脚设置 ADE=0, PFR=1（DDR 将被忽略）。
  所选引脚同时也使用的其他外设功能引脚的所有输出必须关闭。

  示例 2：TIOA11 用作输入引脚时：
  选择 EPFR12:TIOA11E = 00。
  当 TIOA11 接收 TIOA11_0 输入时，选择 EPFR12:TIOA11S = 00 或 01。
  当 TIOA11 接收 TIOA11_1 输入时，选择 EPFR12:TIOA11S = 10。
  当 TIOA11 接收 TIOA11_2 输入时，选择 EPFR12:TIOA11S = 11。
  所选引脚设置 ADE=0, PFR=1（DDR 将被忽略）。
  所选引脚同时也使用的其他外设功能引脚的所有输出必须关闭。

  ** 当引脚被设置为输入时，所选引脚（GPIO、其他外设功能引脚）同时也使用的引脚输出可作为反馈（通过上文未说明的设置）输入至基本计时器。*

4.20 扩展引脚功能设置寄存器 14 (EPFR14)

EPFR14 寄存器设置功能分配至 QPRC 引脚。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	保留							
属性	-							
初始值	-							

位	23	22	21	20	19	18	17	16
字段	保留							
属性	-							
初始值	-							

位	15	14	13	12	11	10	9	8
字段	保留							
属性	-							
初始值	-							

位	7	6	5	4	3	2	1	0
字段	保留		QZIN2S		QBIN2S		QAIN2S	
属性	-		R/W		R/W		R/W	
初始值	-		00		00		00	

寄存器功能

[bit31:6] 保留：保留位

这些位的读取值为 "0"。

写入这些位时，将其设置为 "0"。

[bit5:4] QZIN2S： QPRC-ch.2 ZIN 输入引脚位

选择 QPRC-ch.2 的输入为 ZIN。

位		描述
读取		读取寄存器值。
写入	00	ZIN2_0 用作 ZIN，QPRCch.2 的输入引脚。[初始值]
	01	ZIN2_0 用作 ZIN，QPRCch.2 的输入引脚。
	10	ZIN2_1 用作 ZIN，QPRCch.2 的输入引脚。
	11	ZIN2_2 用作 ZIN，QPRCch.2 的输入引脚。

[bit3:2] QBIN2S： QPRC-ch.2 BIN 输入引脚位

选择 QPRC-ch.2 的输入为 BIN。

位		描述
读取		读取寄存器值。
写入	00	BIN2_0 用作 BIN，QPRC ch.2 的输入引脚。[初始值]
	01	BIN2_0 用作 BIN，QPRCch.2 的输入引脚。
	10	BIN2_1 用作 BIN，QPRCch.2 的输入引脚。
	11	BIN2_2 用作 BIN，QPRCch.2 的输入引脚。

[bit1:0] QAIN2S：QPRC-ch.2 AIN 输入引脚位

选择 QPRC-ch.2 的输入为 AIN。

bit1:0		描述
读取		读取寄存器值。
写入	00	AIN2_0 用作 AIN，QPRC ch.2 的输入引脚。[初始值]
	01	AIN2_0 用作 AIN，QPRCch.2 的输入引脚。
	10	AIN2_1 用作 AIN，QPRCch.2 的输入引脚。
	11	AIN2_2 用作 AIN，QPRCch.2 的输入引脚。

4.21 扩展引脚功能设置寄存器 15 (EPFR15)

EPFR15 寄存器至外部中断引脚的功能分配。

寄存器配置

位	31 30	29 28	27 26	25 24
字段	EINT31S	EINT30S	EINT29S	EINT28S
属性	R/W	R/W	R/W	R/W
初始值	00	00	00	00

位	23 22	21 20	19 18	17 16
字段	EINT27S	EINT26S	EINT25S	EINT24S
属性	R/W	R/W	R/W	R/W
初始值	00	00	00	00

位	15 14	13 12	11 10	9 8
字段	EINT23S	EINT22S	EINT21S	EINT20S
属性	R/W	R/W	R/W	R/W
初始值	00	00	00	00

位	7 6	5 4	3 2	1 0
字段	EINT19S	EINT18S	EINT17S	EINT16S
属性	R/W	R/W	R/W	R/W
初始值	00	00	00	00

寄存器功能

[bit31:30] EINT31S：外部中断输入选择位

选择 EINT31 输入。

bit31:30		描述
读取		读取寄存器值。
写入	00	EINTch.31 输入引脚使用 INT31_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.31 输入引脚使用 INT31_1。
	11	EINTch.31 输入引脚使用 INT31_2。

[bit29:28] EINT30S：外部中断输入选择位

选择 EINT30 输入。

bit29:28		描述
读取		读取寄存器值。
写入	00	EINTch.30 输入引脚使用 INT30_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.30 输入引脚使用 INT30_1。
	11	EINTch.30 输入引脚使用 INT30_2。

[bit27:26] EINT29S：外部中断输入选择位

选择 EINT29 输入。

bit27:26		描述
读取		读取寄存器值。
写入	00	EINTch.29 输入引脚使用 INT29_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.29 输入引脚使用 INT29_1。
	11	EINTch.29 输入引脚使用 INT29_2。

[bit25:24] EINT28S：外部中断输入选择位

选择 EINT28 输入。

bit25:24		描述
读取		读取寄存器值。
写入	00	EINTch.28 输入引脚使用 INT28_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.28 输入引脚使用 INT28_1。
	11	EINTch.28 输入引脚使用 INT28_2。

[bit23:22] EINT27S：外部中断输入选择位

选择 EINT27 输入。

bit23:22		描述
读取		读取寄存器值。
写入	00	EINTch.27 输入引脚使用 INT27_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.27 输入引脚使用 INT27_1。
	11	EINTch.27 输入引脚使用 INT27_2。

[bit21:20] EINT26S：外部中断输入选择位

选择 EINT26 输入。

bit21:20		描述
读取		读取寄存器值。
写入	00	EINTch.26 输入引脚使用 INT26_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.26 输入引脚使用 INT26_1。
	11	EINTch.26 输入引脚使用 INT26_2。

[bit19:18] EINT25S：外部中断输入选择位

选择 EINT25 输入。

bit19:18		描述
读取		读取寄存器值。
写入	00	EINTch.25 输入引脚使用 INT25_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.25 输入引脚使用 INT25_1。
	11	EINTch.25 输入引脚使用 INT25_2。

[bit17:16] EINT24S：外部中断输入选择位

选择 EINT24 输入。

bit17:16		描述
读取		读取寄存器值。
写入	00	EINTch.24 输入引脚使用 INT24_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.24 输入引脚使用 INT24_1。
	11	EINTch.24 输入引脚使用 INT24_2。

[bit15:14] EINT23S：外部中断输入选择位

选择 EINT23 输入。

bit15:14		描述
读取		读取寄存器值。
写入	00	EINTch.23 输入引脚使用 INT23_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.23 输入引脚使用 INT23_1。
	11	EINTch.23 输入引脚使用 INT23_2。

[bit13:12] EINT22S：外部中断输入选择位

选择 EINT22 输入。

bit13:12		描述
读取		读取寄存器值。
写入	00	EINTch.22 输入引脚使用 INT22_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.22 输入引脚使用 INT22_1。
	11	EINTch.22 输入引脚使用 INT22_2。

[bit11:10] EINT21S：外部中断输入选择位

选择 EINT21 输入。

bit11:10		描述
读取		读取寄存器值。
写入	00	EINTch.21 输入引脚使用 INT21_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.21 输入引脚使用 INT21_1。
	11	EINTch.21 输入引脚使用 INT21_2。

[bit9:8] EINT20S：外部中断输入选择位

选择 EINT20 输入。

bit9:8		描述
读取		读取寄存器值。
写入	00	EINTch.20 输入引脚使用 INT20_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.20 输入引脚使用 INT20_1。
	11	EINTch.20 输入引脚使用 INT20_2。

[bit7:6] EINT19S：外部中断输入选择位

选择 EINT19 输入。

bit7:6		描述
读取		读取寄存器值。
写入	00	EINTch.19 输入引脚使用 INT19_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.19 输入引脚使用 INT19_1。
	11	EINTch.19 输入引脚使用 INT19_2。

[bit5:4] EINT18S：外部中断输入选择位

选择 EINT18 输入。

bit5:4		描述
读取		读取寄存器值。
写入	00	EINTch.18 输入引脚使用 INT18_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.18 输入引脚使用 INT18_1。
	11	EINTch.18 输入引脚使用 INT18_2。

[bit3:2] EINT17S：外部中断输入选择位

选择 EINT17 输入。

bit3:2		描述
读取		读取寄存器值。
写入	00	EINTch.17 输入引脚使用 INT17_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.17 输入引脚使用 INT17_1。
	11	EINTch.17 输入引脚使用 INT17_2。

[bit1:0] EINT16S：外部中断输入选择位

选择 EINT16 输入。

bit1:0		描述
读取		读取寄存器值。
写入	00	EINTch.16 输入引脚使用 INT16_0。[初始值]
	01	与写入 00 时相同。
	10	EINTch.16 输入引脚使用 INT16_1。
	11	EINTch.16 输入引脚使用 INT16_2。

4.22 扩展引脚功能设置寄存器 16 (EPFR16)

EPFR16 寄存器分配多功能串口通道 8、通道 9、通道 10 和通道 11 的功能。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	保留				SCK11B		SOT11B	
属性	-				R/W		R/W	
初始值	-				00		00	

位	23	22	21	20	19	18	17	16
字段	SIN11S		SCK10B		SOT10B		SIN10S	
属性	R/W		R/W		R/W		R/W	
初始值	00		00		00		00	

位	15	14	13	12	11	10	9	8
字段	SCK9B		SOT9B		SIN9S		SCK8B	
属性	R/W		R/W		R/W		R/W	
初始值	00		00		00		00	

位	7	6	5	4	3	2	1	0
字段	SOT8B		SIN8S		保留			
属性	R/W		R/W		-			
初始值	00		00		-			

寄存器功能

[bit31:28] 保留：保留位

这些位的读取值为 "0b0000"。

写入这些位时，将其设置为 "0b0000"。

[bit27:26] SCK11B： SCK11 输入/输出选择位

选择 SCK11 输入/输出。

bit27:26		描述
读取		读取寄存器值。
写入	00	MFSch.11 SCK 输入引脚使用 SCK11_0。 不产生输出。[初始值]
	01	MFSch.11 SCK 输入引脚使用 SCK11_0。 输出引脚使用 SCK11_0。
	10	MFSch.11 SCK 输入引脚使用 SCK11_1。 输出引脚使用 SCK11_1。
	11	MFSch.11 SCK 输入引脚使用 SCK11_2。 输出引脚使用 SCK11_2。

[bit25:24] SOT11B：SOT11 输入/输出选择位

选择 SOT11 输入/输出。

bit25:24		描述
读取		读取寄存器值。
写入	00	MFSch.11 SOT 输入引脚使用 SOT11_0。 不产生输出。[初始值]
	01	MFSch.11 SOT 输入引脚使用 SOT11_0。 输出引脚使用 SOT11_0。
	10	MFSch.11 SOT 输入引脚使用 SOT11_1。 输出引脚使用 SOT11_1。
	11	MFSch.11 SOT 输入引脚使用 SOT11_2。 输出引脚使用 SOT11_2。

[bit23:22] SIN11S：SIN11 输入选择位

选择 SIN11 输入。

bit23:22		描述
读取		读取寄存器值。
写入	00	MFSch.11 SIN 输入引脚使用 SIN11_0。[初始值]
	01	与写入 00 时相同。
	10	MFSch.11 SIN 输入引脚使用 SIN11_1。
	11	MFSch.11 SIN 输入引脚使用 SIN11_2。

[bit21:20] SCK10B： SCK10 输入/输出选择位

选择 SCK10 输入/输出。

bit21:20		描述
读取		读取寄存器值。
写入	00	MFSch.10 SCK 输入引脚使用 SCK10_0。 不产生输出。[初始值]
	01	MFSch.10 SCK 输入引脚使用 SCK10_0。 输出引脚使用 SCK10_0。
	10	MFSch.10 SCK 输入引脚使用 SCK10_1。 输出引脚使用 SCK10_1。
	11	MFSch.10 SCK 输入引脚使用 SCK10_2。 输出引脚使用 SCK10_2。

[bit19:18] SOT10B：SOT10 输入/输出选择位

选择 SOT10 输入/输出。

bit19:18		描述
读取		读取寄存器值。
写入	00	MFSch.10 SOT 输入引脚使用 SOT10_0。 不产生输出。[初始值]
	01	MFSch.10 SOT 输入引脚使用 SOT10_0。 输出引脚使用 SOT10_0。
	10	MFSch.10 SOT 输入引脚使用 SOT10_1。 输出引脚使用 SOT10_1。
	11	MFSch.10 SOT 输入引脚使用 SOT10_2。 输出引脚使用 SOT10_2。

[bit17:16] SIN10S：SIN10 输入选择位

选择 SIN10 输入。

bit17:16		描述
读取		读取寄存器值。
写入	00	MFSch.10 SIN 输入引脚使用 SIN10_0。[初始值]
	01	与写入 00 时相同。
	10	MFSch.10 SIN 输入引脚使用 SIN10_1。
	11	MFSch.10 SIN 输入引脚使用 SIN10_2。

[bit15:14] SCK9B： SCK9 输入/输出选择位

选择 SCK9 输入/输出。

bit15:14		描述
读取		读取寄存器值。
写入	00	MFSch.9 SCK 输入引脚使用 SCK9_0。 不产生输出。[初始值]
	01	MFSch.9 SCK 输入引脚使用 SCK9_0。 输出引脚使用 SCK9_0。
	10	MFSch.9 SCK 输入引脚使用 SCK9_1。 输出引脚使用 SCK9_1。
	11	MFSch.9 SCK 输入引脚使用 SCK9_2。 输出引脚使用 SCK9_2。

[bit13:12] SOT9B：SOT9 输入/输出选择位

选择 SOT9 输入/输出。

bit13:12		描述
读取		读取寄存器值。
写入	00	MFSch.9 SOT 输入引脚使用 SOT9_0。 不产生输出。[初始值]
	01	MFSch.9 SOT 输入引脚使用 SOT9_0。 输出引脚使用 SOT9_0。
	10	MFSch.9 SOT 输入引脚使用 SOT9_1。 输出引脚使用 SOT9_1。
	11	MFSch.9 SOT 输入引脚使用 SOT9_2。 输出引脚使用 SOT9_2。

[bit11:10] SIN9S：SIN9 输入选择位

选择 SIN9 输入。

bit11:10		描述
读取		读取寄存器值。
写入	00	MFSch.9 SIN 输入引脚使用 SIN9_0。[初始值]
	01	与写入 00 时相同。
	10	MFSch.9 SIN 输入引脚使用 SIN9_1。
	11	MFSch.9 SIN 输入引脚使用 SIN9_2。

[bit9:8] SCK8B：SCK8 输入/输出选择位

选择 SCK8 输入/输出。

bit9:8		描述
读取		读取寄存器值。
写入	00	MFSch.8 SCK 输入引脚使用 SCK8_0。 不产生输出。[初始值]
	01	MFSch.8 SCK 输入引脚使用 SCK8_0。 输出引脚使用 SCK8_0。
	10	MFSch.8 SCK 输入引脚使用 SCK8_1。 输出引脚使用 SCK8_1。
	11	MFSch.8 SCK 输入引脚使用 SCK8_2。 输出引脚使用 SCK8_2。

[bit7:6] SOT8B：SOT8 输入/输出选择位

选择 SOT8 输入/输出。

bit7:6		描述
读取		读取寄存器值。
写入	00	MFSch.8 SOT 输入引脚使用 SOT8_0。 不产生输出。[初始值]
	01	MFSch.8 SOT 输入引脚使用 SOT8_0。 输出引脚使用 SOT8_0。
	10	MFSch.8 SOT 输入引脚使用 SOT8_1。 输出引脚使用 SOT8_1。
	11	MFSch.8 SOT 输入引脚使用 SOT8_2。 输出引脚使用 SOT8_2。

[bit5:4] SIN8S：SIN8 输入选择位

选择 SIN4 输入。

bit5:4		描述
读取		读取寄存器值。
写入	00	MFSch.8 SIN 输入引脚使用 SIN8_0。[初始值]
	01	与写入 00 时相同。
	10	MFSch.8 SIN 输入引脚使用 SIN8_1。
	11	MFSch.8 SIN 输入引脚使用 SIN8_2。

[bit3:0] 保留：保留位

这些位的读取值为 "0b0000"。

写入这些位时，将其设置为 "0b0000"。

4.23 扩展引脚功能设置寄存器 17 (EPFR17)

EPFR17 寄存器分配多功能串口通道 12、通道 13、通道 14 和通道 15 的功能。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	保留				SCK15B		SOT15B	
属性	-				R/W		R/W	
初始值	-				00		00	

位	23	22	21	20	19	18	17	16
字段	SIN15S		SCK14B		SOT14B		SIN14S	
属性	R/W		R/W		R/W		R/W	
初始值	00		00		00		00	

位	15	14	13	12	11	10	9	8
字段	SCK13B		SOT13B		SIN13S		SCK12B	
属性	R/W		R/W		R/W		R/W	
初始值	00		00		00		00	

位	7	6	5	4	3	2	1	0
字段	SOT12B		SIN12S		保留			
属性	R/W		R/W		-			
初始值	00		00		-			

寄存器功能

[bit31:28] 保留：保留位

这些位的读取值为 "0b0000"。

写入这些位时，将其设置为 "0b0000"。

[bit27:26] SCK15B： SCK15 输入/输出选择位

选择 SCK15 输入/输出。

bit27:26		描述
读取		读取寄存器值。
写入	00	MFSch.15 SCK 输入引脚使用 SCK15_0。 不产生输出。[初始值]
	01	MFSch.15 SCK 输入引脚使用 SCK15_0。 输出引脚使用 SCK15_0。
	10	MFSch.15 SCK 输入引脚使用 SCK15_1。 输出引脚使用 SCK15_1。
	11	MFSch.15 SCK 输入引脚使用 SCK15_2。 输出引脚使用 SCK15_2。

[bit25:24] SOT15B：SOT15 输入/输出选择位

选择 SOT15 输入/输出。

bit25:24		描述
读取		读取寄存器值。
写入	00	MFSch.15 SOT 输入引脚使用 SOT15_0。 不产生输出。[初始值]
	01	MFSch.15 SOT 输入引脚使用 SOT15_0。 输出引脚使用 SOT15_0。
	10	MFSch.15 SOT 输入引脚使用 SOT15_1。 输出引脚使用 SOT15_1。
	11	MFSch.15 SOT 输入引脚使用 SOT15_2。 输出引脚使用 SOT15_2。

[bit23:22] SIN15S：SIN15 输入选择位

选择 SIN15 输入。

bit23:22		描述
读取		读取寄存器值。
写入	00	MFSch.15 SIN 输入引脚使用 SIN15_0。[初始值]
	01	与写入 00 时相同。
	10	MFSch.15 SIN 输入引脚使用 SIN15_1。
	11	MFSch.15 SIN 输入引脚使用 SIN15_2。

[bit21:20] SCK14B： SCK14 输入/输出选择位

选择 SCK14 输入/输出。

bit21:20		描述
读取		读取寄存器值。
写入	00	MFSch.14 SCK 输入引脚使用 SCK14_0。 不产生输出。[初始值]
	01	MFSch.14 SCK 输入引脚使用 SCK14_0。 输出引脚使用 SCK14_0。
	10	MFSch.14 SCK 输入引脚使用 SCK14_1。 输出引脚使用 SCK14_1。
	11	MFSch.14 SCK 输入引脚使用 SCK14_2。 输出引脚使用 SCK14_2。

[bit19:18] SOT14B：SOT14 输入/输出选择位

选择 SOT14 输入/输出。

bit19:18		描述
读取		读取寄存器值。
写入	00	MFSch.14 SOT 输入引脚使用 SOT14_0。 不产生输出。[初始值]
	01	MFSch.14 SOT 输入引脚使用 SOT14_0。 输出引脚使用 SOT14_0。
	10	MFSch.14 SOT 输入引脚使用 SOT14_1。 输出引脚使用 SOT14_1。
	11	MFSch.14 SOT 输入引脚使用 SOT14_2。 输出引脚使用 SOT14_2。

[bit17:16] SIN14S：SIN14 输入选择位

选择 SIN14 输入。

bit17:16		描述
读取		读取寄存器值。
写入	00	MFSch.14 SIN 输入引脚使用 SIN14_0。[初始值]
	01	与写入 00 时相同。
	10	MFSch.14 SIN 输入引脚使用 SIN14_1。
	11	MFSch.14 SIN 输入引脚使用 SIN14_2。

[bit15:14] SCK13B：SCK13 输入/输出选择位

选择 SCK13 输入/输出。

bit15:14		描述
读取		读取寄存器值。
写入	00	MFSch.13 SCK 输入引脚使用 SCK13_0。 不产生输出。[初始值]
	01	MFSch.13 SCK 输入引脚使用 SCK13_0。 输出引脚使用 SCK13_0。
	10	MFSch.13 SCK 输入引脚使用 SCK13_1。 输出引脚使用 SCK13_1。
	11	MFSch.13 SCK 输入引脚使用 SCK13_2。 输出引脚使用 SCK13_2。

[bit13:12] SOT13B：SOT13 输入/输出选择位

选择 SOT13 输入/输出。

bit13:12		描述
读取		读取寄存器值。
写入	00	MFSch.13 SOT 输入引脚使用 SOT13_0。 不产生输出。[初始值]
	01	MFSch.13 SOT 输入引脚使用 SOT13_0。 输出引脚使用 SOT13_0。
	10	MFSch.13 SOT 输入引脚使用 SOT13_1。 输出引脚使用 SOT13_1。
	11	MFSch.13 SOT 输入引脚使用 SOT13_2。 输出引脚使用 SOT13_2。

[bit11:10] SIN13S：SIN13 输入选择位

选择 SIN13 输入。

bit11:10		描述
读取		读取寄存器值。
写入	00	MFSch.13 SIN 输入引脚使用 SIN13_0。[初始值]
	01	与写入 00 时相同。
	10	MFSch.13 SIN 输入引脚使用 SIN13_1。
	11	MFSch.13 SIN 输入引脚使用 SIN13_2。

[bit9:8] SCK12B：SCK12 输入/输出选择位

选择 SCK12 输入/输出。

bit9:8		描述
读取		读取寄存器值。
写入	00	MFSch.12 SCK 输入引脚使用 SCK12_0。 不产生输出。[初始值]
	01	MFSch.12 SCK 输入引脚使用 SCK12_0。 输出引脚使用 SCK12_0。
	10	MFSch.12 SCK 输入引脚使用 SCK12_1。 输出引脚使用 SCK12_1。
	11	MFSch.12 SCK 输入引脚使用 SCK12_2。 输出引脚使用 SCK12_2。

[bit7:6] SOT12B： SOT12 输入/输出选择位

选择 SOT12 输入/输出。

bit7:6		描述
读取		读取寄存器值。
写入	00	MFSch.12 SOT 输入引脚使用 SOT12_0。 不产生输出。[初始值]
	01	MFSch.12 SOT 输入引脚使用 SOT12_0。 输出引脚使用 SOT12_0。
	10	MFSch.12 SOT 输入引脚使用 SOT12_1。 输出引脚使用 SOT12_1。
	11	MFSch.12 SOT 输入引脚使用 SOT12_2。 输出引脚使用 SOT12_2。

[bit5:4] SIN12S：SIN12 输入选择位

选择 SIN12 输入。

bit5:4		描述
读取		读取寄存器值。
写入	00	MFSch.12 SIN 输入引脚使用 SIN12_0。[初始值]
	01	与写入 00 时相同。
	10	MFSch.12 SIN 输入引脚使用 SIN12_1。
		MFSch.12 SIN 输入引脚使用 SIN12_2。

[bit3:0] 保留：保留位

这些位的读取值为 "0b0000"。

写入这些位时，将其设置为 "0b0000"。

4.24 扩展引脚功能设置寄存器 18 (EPFR18)

EPFR18 寄存器分配 HDMI-CEC 引脚的功能。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	保留							
属性	-							
初始值	-							

位	23	22	21	20	19	18	17	16
字段	保留							
属性	-							
初始值	-							

位	15	14	13	12	11	10	9	8
字段	保留							
属性	-							
初始值	-							

位	7	6	5	4	3	2	1	0
字段	保留				CECR1B		CECR0B	
属性	-				R/W		R/W	
初始值	-				00		00	

寄存器功能

[bit31:4] 保留：保留位

这些位的读取值为 "0"。

写入这些位时，将其设置为 "0"。

[bit3:2] CECR1B： CEC1 输入/输出选择位

此位为 HDMI-CEC/遥控接收 ch.1 的 I/O 引脚 CEC1 选择 I/O。

bit3:2		描述
读取		读取寄存器值。
写入	00	HDMI-CEC/遥控接收 ch.1 未输入/输出。 [初始值]
	01	使用 CEC1_0，HDMI-CEC/遥控接收 ch.1 输入/输出。
	10	使用 CEC1_1，HDMI-CEC/遥控接收 ch.1 输入/输出。
	11	禁止设置。

注意事项：

- *在 TYPE 3-M0+ 产品中，CECR1B 不工作。使用 CEC_CTL 寄存器代替。参见低功耗章节中的“寄存器”部分。写入这些位时，将其设置为 "00"。*

[bit1:0] CECR0B： CEC0 输入/输出选择位

此位为 HDMI-CEC/遥控接收 ch.0 的 I/O 引脚 CEC0 选择 I/O。

bit1:0	描述
读取	读取寄存器值。
写入 00	HDMI-CEC/遥控接收 ch.0 未输入/输出。 [初始值]
写入 01	使用 CEC0_0，HDMI-CEC/遥控接收 ch.0 输入/输出。
写入 10	使用 CEC0_1，HDMI-CEC/遥控接收 ch.0 输入/输出。
写入 11	禁止设置。

注意事项：

- *在 TYPE 3-M0+ 产品中，CECR0B 不工作。使用 CEC_CTL 寄存器代替。参见低功耗章节中的“寄存器”部分。写入这些位时，将其设置为 "00"。*

4.25 扩展引脚功能设置寄存器 21 (EPFR21)

EPFR21 寄存器设置 QPRC 的功能分配。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	保留							
属性	-							
初始值	-							

位	23	22	21	20	19	18	17	16
字段	保留							
属性	-							
初始值	-							

位	15	14	13	12	11	10	9	8
字段	保留							
属性	-							
初始值	-							

位	7	6	5	4	3	2	1	0
字段	保留					QZIN0S[2]	QBIN0S[2]	QAIN0S[2]
属性	-					R/W	R/W	R/W
初始值	-					0	0	0

寄存器功能

[bit31:3] 保留：保留位

这些位的读取值为 "0"。

写入这些位时，将其设置为 "0"。

[bit2] QZIN0S[2]： QPRC-ch.0ZIN 输入引脚位

选择 QPRC-ch.0 的输入为 ZIN。如果使用 ZIN0_3，设置此位以及扩展引脚功能设置寄存器 9 (EPFR09) 中的 QZIN0S 位。

Bit[2],EPFR09[5:4]		描述
读取		读取寄存器值。
写入	000	ZIN0_0 用作 ZIN，QDU ch.0 的输入引脚。 [初始值]
	001	ZIN0_0 用作 ZIN，QDU ch.0 的输入引脚。
	010	ZIN0_1 用作 ZIN，QDU ch.0 的输入引脚。
	011	ZIN0_2 用作 ZIN，QDU ch.0 的输入引脚。
	100	ZIN0_3 用作 ZIN，QDU ch.0 的输入引脚。
	其它	禁止设置。

[bit1] QBIN0S[2]：QPRC-ch.0BIN 输入引脚位

选择 QPRC-ch.0 的输入为 BIN。如果使用 BIN0_3，设置此位以及扩展引脚功能设置寄存器 9 (EPFR09) 中的 QBIN0S 位。

Bit[1],EPFR09[3:2]		描述
读取		读取寄存器值。
写入	000	BIN0_0 用作 BIN，QDU ch.0 的输入引脚。 [初始值]
	001	BIN0_0 用作 BIN，QDU ch.0 的输入引脚。
	010	BIN0_1 用作 BIN，QDU ch.0 的输入引脚。
	011	BIN0_2 用作 BIN，QDU ch.0 的输入引脚。
	100	BIN0_3 用作 BIN，QDU ch.0 的输入引脚。
	其它	禁止设置。

[bit0] QAIN0S[2]：QPRC-ch.0AIN 输入引脚位

选择 QPRC-ch.0 的输入为 AIN。如果使用 AIN0_3，设置此位以及扩展引脚功能设置寄存器 9 (EPFR09) 中的 QAIN0S 位。

Bit[0],EPFR09[1:0]		描述
读取		读取寄存器值。
写入	000	AIN0_0 用作 AIN，QDU ch.0 的输入引脚。 [初始值]
	001	AIN0_0 用作 AIN，QDU ch.0 的输入引脚。
	010	AIN0_1 用作 AIN，QDU ch.0 的输入引脚。
	011	AIN0_2 用作 AIN，QDU ch.0 的输入引脚。
	100	AIN0_3 用作 AIN，QDU ch.0 的输入引脚。
	其它	禁止设置。

4.26 扩展引脚功能设置寄存器 22 (EPFR22)

EPFR22 寄存器分配多功能串口通道 0、通道 1、通道 2、通道 3 的功能。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	保留							
属性	-							
初始值	-							

位	23	22	21	20	19	18	17	16
字段	保留							
属性	-							
初始值	-							

位	15	14	13	12	11	10	9	8
字段	SCS31E		SCS30B		保留			
R/W	R/W		R/W		R/W			
00	00		00		00			

位	7	6	5	4	3	2	1	0
字段	SCS11E		SCS10B		保留			
属性	R/W		R/W		-			
初始值	00		00		-			

寄存器功能

[bit31:16] 保留：保留位

这些位的读取值为 "0"。

写入这些位时，将其设置为 "0"。

[bit15:14] SCS31E： SCS31 输出选择位

选择 SCS31 输出。

bit15:14		描述
读取		读取寄存器值。
写入	00	不产生 MFS ch.3 SCS31 的输出。[初始值]
	01	MFS ch. 3 SCS31 输出引脚使用 SCS31_0。
	10	MFS ch. 3 SCS31 输出引脚使用 SCS31_1。
	11	MFS ch. 3 SCS31 输出引脚使用 SCS31_2。

[bit13:12] SCS30B： SCS30 输入/输出选择位

选择 SCS30 输入/输出。

bit13:12		描述
读取		读取寄存器值。
写入	00	MFS ch.3 SCS30 输入引脚使用 SCS30_0。 不产生输出。[初始值]
	01	MFS ch.3 SCS30 输入引脚使用 SCS30_0。 输出引脚使用 SCS30_0。
	10	MFS ch.3 SCS30 输入引脚使用 SCS30_1。 输出引脚使用 SCS30_1。
	11	MFS ch.3 SCS30 输入引脚使用 SCS30_2。 输出引脚使用 SCS30_2。

[bit11:8] 保留：保留位

这些位的读取值为 "0"。

写入这些位时，将其设置为 "0"。

[bit7:6] SCS11E： SCS11 输出选择位

选择 SCS11 输出。

bit7:6		描述
读取		读取寄存器值。
写入	00	不产生 MFS ch.1 SCS11 的输出。[初始值]
	01	MFS ch.1 SCS11 输出引脚使用 SCS11_0。
	10	MFS ch.1 SCS11 输出引脚使用 SCS11_1。
	11	MFS ch.1 SCS11 输出引脚使用 SCS11_2。

[bit5:4] SCS10B： SCS10 输入/输出选择位

选择 SCS10 输入/输出。

bit5:4		描述
读取		读取寄存器值。
写入	00	MFS ch.1 SCS10 输入引脚使用 SCS10_0。 不产生输出。[初始值]
	01	MFS ch.1 SCS10 输入引脚使用 SCS10_0。 输出引脚使用 SCS10_0。
	10	MFS ch.1 SCS10 输入引脚使用 SCS10_1。 输出引脚使用 SCS10_1。
	11	MFS ch.1 SCS10 输入引脚使用 SCS10_2。 输出引脚使用 SCS10_2。

[bit3:0] 保留：保留位

这些位的读取值为 "0"。

写入这些位时，将其设置为 "0"。

4.27 扩展引脚功能设置寄存器 23 (EPFR23)

EPFR23 寄存器分配多功能串口 ch.6 至 ch.7 的功能。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	保留							
属性	-							
初始值	-							

位	23	22	21	20	19	18	17	16
字段	保留							
属性	-							
初始值	-							

位	15	14	13	12	11	10	9	8
字段	SCS73E		SCS72E		SCS71E		SCS70B	
属性	R/W		R/W		R/W		R/W	
初始值	00		00		00		00	

位	7	6	5	4	3	2	1	0
字段	SCS63E		SCS62E		SCS61E		SCS60B	
属性	R/W		R/W		R/W		R/W	
初始值	00		00		00		00	

寄存器功能

[bit31:16] 保留：保留位

这些位的读取值为 "0x0000"。

写入这些位时，将其设置为 "0x0000"。

[bit15:14] SCS73E：SCS73 输出选择位

选择 SCS73 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 MFS ch7 SCS73 的输出。[初始值]
	01	MFS ch.7 SCS73 输出引脚使用 SCS73_0。
	10	MFS ch.7 SCS73 输出引脚使用 SCS73_1。
	11	MFS ch.7 SCS73 输出引脚使用 SCS73_2。

[bit13:12] SCS72E：SCS72 输出选择位

选择 SCS72 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 MFS ch7 SCS72 的输出。[初始值]
	01	MFS ch.7 SCS72 输出引脚使用 SCS72_0。
	10	MFS ch.7 SCS72 输出引脚使用 SCS72_1。
	11	MFS ch.7 SCS72 输出引脚使用 SCS72_2。

[bit11:10] SCS71E： SCS71 输出选择位

选择 SCS71 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 MFS ch7 SCS71 的输出。[初始值]
	01	MFS ch.7 SCS71 输出引脚使用 SCS71_0。
	10	MFS ch.7 SCS71 输出引脚使用 SCS71_1。
	11	MFS ch.7 SCS71 输出引脚使用 SCS71_2。

[bit9:8] SCS70B： SCS70 输入/输出选择位

选择 SCS70 输入/输出。

位		描述
读取		读取寄存器值。
写入	00	MFS ch.7 SCS70 输入引脚使用 SCS70_0。 不产生输出。[初始值]
	01	MFS ch.7 SCS70 输入引脚使用 SCS70_0。 输出引脚使用 SCS70_0。
	10	MFS ch.7 SCS70 输入引脚使用 SCS70_1。 输出引脚使用 SCS70_1。
	11	MFS ch.7 SCS70 输入引脚使用 SCS70_2。 输出引脚使用 SCS70_2。

[bit7:6] SCS63E： SCS63 输出选择位

选择 SCS63 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 MFS ch6 SCS63 的输出。[初始值]
	01	MFS ch.6 SCS63 输出引脚使用 SCS63_0。
	10	MFS ch.6 SCS63 输出引脚使用 SCS63_1。
	11	MFS ch.6 SCS63 输出引脚使用 SCS63_2。

[bit5:4] SCS62E： SCS62 输出选择位

选择 SCS62 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 MFS ch6 SCS62 的输出。[初始值]
	01	MFS ch.6 SCS62 输出引脚使用 SCS62_0。
	10	MFS ch.6 SCS62 输出引脚使用 SCS62_1。
	11	MFS ch.6 SCS62 输出引脚使用 SCS62_2。

[bit3:2] SCS61E： SCS61 输出选择位

选择 SCS61 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 MFS ch6 SCS61 的输出。[初始值]
	01	MFS ch.6 SCS61 输出引脚使用 SCS61_0。
	10	MFS ch.6 SCS61 输出引脚使用 SCS61_1。
	11	MFS ch.6 SCS61 输出引脚使用 SCS61_2。

[bit1:0] SCS60B： SCS60 输入/输出选择位

选择 SCS60 输入/输出。

位		描述
读取		读取寄存器值。
写入	00	MFS ch.6 SCS60 输入引脚使用 SCS60_0。 不产生输出。[初始值]
	01	MFS ch.6 SCS60 输入引脚使用 SCS60_0。 输出引脚使用 SCS60_0。
	10	MFS ch.6 SCS60 输入引脚使用 SCS60_1。 输出引脚使用 SCS60_1。
	11	MFS ch.6 SCS60 输入引脚使用 SCS60_2。 输出引脚使用 SCS60_2。

4.28 扩展引脚功能设置寄存器 31 (EPFR31)

EPFR31 寄存器分配 I2C 从机唤醒的功能。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	保留				SI2CSDA6B		SI2CSCL6B	
属性		R/W			R/W		R/W	
初始值		0000			00		00	

位	23	22	21	20	19	18	17	16
字段	保留							
属性				R/W				
初始值				00000000				

位	15	14	13	12	11	10	9	8
字段	保留							
属性				R/W				
初始值				00000000				

位	7	6	5	4	3	2	1	0
字段	保留							
属性				R/W				
初始值				00000000				

寄存器功能

[bit31:28] 保留：保留位

这些位的读取值为 "0x0"。

写入这些位时，将其设置为 "0x0"。

[bit27:26] SI2CSDA6B： I2C 从机 ch.6 SDA 引脚选择位

选择 ch.6 SDA 引脚。

位		描述
读取		读取寄存器值。
写入	00	I2CSLAVE ch.6 SDA 输入引脚使用 SI2CSDA6_0。 不产生输出。[初始值]
	01	I2CSLAVE ch.6 SDA 输入引脚使用 SI2CSDA6_0。 输出引脚使用 SI2CSDA6_0。
	10	I2CSLAVE ch.6 SDA 输入引脚使用 SI2CSDA6_1。 输出引脚使用 SI2CSDA6_1。
	11	I2CSLAVE ch.6 SDA 输入引脚使用 SI2CSDA6_2。 输出引脚使用 SI2CSDA6_2。

[bit25:24] SI2CSCL6B： I2C 从机 ch.6 SCL 引脚选择位

选择 ch.6 SCL 引脚。

位		描述
读取		读取寄存器值。
写入	00	I2CSLAVE ch.6 SCL 输入引脚使用 SI2CSCL6_0。 不产生输出。[初始值]
	01	I2CSLAVE ch.6 SCL 输入引脚使用 SI2CSCL6_0。 输出引脚使用 SI2CSCL6_0。
	10	I2CSLAVE ch.6 SCL 输入引脚使用 SI2CSCL6_1。 输出引脚使用 SI2CSCL6_1。
	11	I2CSLAVE ch.6 SCL 输入引脚使用 SI2CSCL6_2。 输出引脚使用 SI2CSCL6_2。

[bit23:0] 保留：保留位

这些位的读取值为 "0x000000"。

写入这些位时，将其设置为 "0x000000"。

4.29 扩展引脚功能设置寄存器 33 (EPFR33)

EPFR33 寄存器分配智能卡的功能

寄存器配置

位	31	30	29	28	27	26	25	24
字段	保留		保留		CLK1E		VCC1E	
属性	-		-		R/W		R/W	
初始值	-		-		00		00	

位	23	22	21	20	19	18	17	16
字段	VPEN1E		RST1E		DATA1B		CIN1S	
属性	R/W		R/W		R/W		R/W	
初始值	00		00		00		00	

位	15	14	13	12	11	10	9	8
字段	保留		保留		CLK0E		VCC0E	
属性	-		-		R/W		R/W	
初始值	-		-		00		00	

位	7	6	5	4	3	2	1	0
字段	VPEN0E		RST0E		DATA0B		CIN0S	
属性	R/W		R/W		R/W		R/W	
初始值	00		00		00		00	

寄存器功能

[bit31:28] 保留：保留位

这些位的读取值为 "0x0000"。

写入这些位时，将其设置为 "0x0000"。

[bit27:26] CLK1E： CLK1 输出选择位

选择 ICC CLK1 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 ICC CLK1 的输出。[初始值]
	01	ICC CLK1 输出引脚使用 CLK1_0。
	10	ICC CLK1 输出引脚使用 CLK1_1。
	11	禁止设置。

[bit25:24] VCC1E：VCC1 输出选择位

选择 VCC1 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 VCC1 的输出。[初始值]
	01	VCC1 输出引脚使用 VCC1_0。
	10	VCC1 输出引脚使用 VCC1_1。
	11	禁止设置。

[bit23:22] VPEN1E：VCC1 输出选择位

选择 ICC VPEN1 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 ICC VPEN1 的输出。[初始值]
	01	ICC VPEN1 输出引脚使用 VPEN1_0。
	10	ICC VPEN1 输出引脚使用 VPEN1_1。
	11	禁止设置。

[bit21:20] RST1E：VCC1 输出选择位

选择 ICC RST1 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 ICC RST1 的输出。[初始值]
	01	ICC RST1 输出引脚使用 RST1_0。
	10	ICC RST1 输出引脚使用 RST1_1。
	11	禁止设置。

[bit19:18] DATA1B：DATA1 总线选择位

选择 DATA1 输入/输出。

位		描述
读取		读取寄存器值。
写入	00	ICC DATA1 输入引脚使用 DATA1_0，禁止 DATA1_0 输出。[初始值]
	01	与写入 00 时相同。
	10	ICC DATA1 输入/输出引脚使用 DATA1_1。
	11	禁止设置。

[bit17:16] CIN1S：CIN1 输入选择位

选择 CIN1 卡插入检测输入。

位		描述
读取		读取寄存器值。
写入	00	ICC CIN1 输入引脚使用 CIN1_0。[初始值]
	01	与写入 00 时相同。
	10	ICC CIN1 输入引脚使用 CIN1_1。
	11	不产生 CIN1 的输入。

[bit11:10] CLK0E：CLK0 输出选择位

选择 ICC CLK0 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 ICC CLK0 的输出。[初始值]
	01	ICC CLK0 输出引脚使用 CLK0_0。
	10	ICC CLK0 输出引脚使用 CLK0_1。
	11	禁止设置。

[bit9:8] VCC0E：VCC0 输出选择位

选择 VCC0 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 VCC0 的输出。[初始值]
	01	VCC0 输出引脚使用 VCC0_0。
	10	VCC0 输出引脚使用 VCC0_1。
	11	禁止设置。

[bit7:6] VPEN0E：VPEN0 输出选择位

选择 ICC VPEN0 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 ICC VPEN0 的输出。[初始值]
	01	ICC VPEN0 输出引脚使用 VPEN0_0。
	10	ICC VPEN0 输出引脚使用 VPEN0_1。
	11	禁止设置。

[bit5:4] RST0E：RST0 输出选择位

选择 ICC RST0 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 ICC RST0 的输出。[初始值]
	01	ICC RST0 输出引脚使用 RST0_0。
	10	ICC RST0 输出引脚使用 RST0_1。
	11	禁止设置。

[bit3:2] DATA0B：DATA0 总线选择位

选择 DATA0 输入/输出。

位		描述
读取		读取寄存器值。
写入	00	ICC DATA0 输入/输出引脚使用 DATA0_0。[初始值]
	01	与写入 00 时相同。
	10	ICC DATA0 输入/输出引脚使用 DATA0_1。
	11	禁止设置。

[bit1:0] CIN0S：CIN0 输入选择位

选择 CIN0 卡插入检测输入。

位		描述
读取		读取寄存器值。
写入	00	ICC CIN0 输入引脚使用 CIN0_0。[初始值]
	01	与写入 00 时相同。
	10	ICC CIN0 输入引脚使用 CIN0_1。
	11	不产生 CIN1 的输入。

4.30 扩展引脚功能设置寄存器 34 (EPFR34)

EPFR34 寄存器分配 MFS 流控制功能。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	保留							
属性	-							
初始值	-							

位	23	22	21	20	19	18	17	16
字段	保留							
属性	-							
初始值	-							

位	15	14	13	12	11	10	9	8
字段	保留							
属性	-							
初始值	-							

位	7	6	5	4	3	2	1	0
字段	CTS1S		RTS1E		保留		保留	
属性	R/W		R/W		-		-	
初始值	00		00		-		-	

寄存器功能

[bit31:8] 保留：保留位

这些位的读取值为 "0x0"。

写入这些位时，将其设置为 "0x0"。

[bit7:6] CTS1SCTS1 输入选择位

选择 CTS1 输入。

位		描述
读取		读取寄存器值。
写入	00	不产生 MFSch.1CTS 输入引脚。[初始值]
	01	与写入 00 时相同。
	10	MFSch.1CTS 输入引脚使用 CTS1_1。
	11	MFSch.1CTS 输入引脚使用 CTS1_2。

[bit5:4] RTS1E：RTS1 输出选择位

选择 RTS1 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 MFSch.1RTS 的输出。[初始值]
	01	与写入 00 时相同。
	10	MFSch.1RTS 输出引脚使用 RTS1_1。
	11	MFSch.1RTS 输出引脚使用 RTS1_2。

[bit3:0] 保留：保留位

这些位的读取值为 "0x0"。

写入这些位时，将其设置为 "0x0"。

4.31 扩展引脚功能设置寄存器 37 (EPFR37)

EPFR37 寄存器分配 MFS ch.4、ch.5 的 MFSI2S 功能。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	保留		保留		SDO5E		SDI5S	
属性	-		-		R/W		R/W	
初始值	-		-		00		00	

位	23	22	21	20	19	18	17	16
字段	WS5B		SCK5B		MCK5E		MCK5S	
属性	R/W		R/W		R/W		R/W	
初始值	00		00		00		00	

位	15	14	13	12	11	10	9	8
字段	保留		保留		SDO4E		SDI4S	
属性	-		-		R/W		R/W	
初始值	-		-		00		00	

位	7	6	5	4	3	2	1	0
字段	WS4B		SCK4B		MCK4E		MCK4S	
属性	R/W		R/W		R/W		R/W	
初始值	00		00		00		00	

寄存器功能

[bit31:28] 保留：保留位

这些位的读取值为 "0x0"。

写入这些位时，将其设置为 "0x0"。

[bit27:26] SDO5E： MFSI2S SDO5 输出选择位

选择 I2SDO5 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 I2S ch.5I2SDO 的输出。[初始值]
	01	I2S ch.5I2SDO 输出引脚使用 I2SDO5_0。
	10	I2S ch.5I2SDO 输出引脚使用 I2SDO5_1。
	11	I2S ch.5I2SDO 输出引脚使用 I2SDO5_2。

[bit25:24] SDI5S：MFSI2S I2SDI5 输入选择位

选择 I2SDI5 输入。

位		描述
读取		读取寄存器值。
写入	00	不产生 I2S ch.5I2SDI 的输入。[初始值]
	01	I2S ch.5 I2SDI 输入引脚使用 I2SDI5_0。
	10	I2S ch.5 I2SDI 输入引脚使用 I2SDI5_1。
	11	I2S ch.5 I2SDI 输入引脚使用 I2SDI5_2。

[bit23:22] MFSI2S WS5B： I2SWS5 输出选择位

选择 I2SWS5 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 I2S ch.5 I2SWS 的输出。[初始值]
	01	I2S ch.5 I2SWS 输出引脚使用 I2SWS5_0。
	10	I2S ch.5 I2SWS 输出引脚使用 I2SWS5_1。
	11	I2S ch.5 I2SWS 输出引脚使用 I2SWS5_2。

[bit21:20] MFSI2SSCK5B：I2SCK5 输出选择位

选择 I2SCK5 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 I2S ch.5 I2SCK 的输出。[初始值]
	01	I2S ch.5 I2SCK 输出引脚使用 I2SCK5_0。
	10	I2S ch.5 I2SCK 输出引脚使用 I2SCK5_1。
	11	I2S ch.5 I2SCK 输出引脚使用 I2SCK5_2。

[bit19:18] MFSI2SMCLK5E：I2SMCLK5 输出选择位

选择 I2SMCLK5 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 I2S ch.5 I2SMCLK 的输出。[初始值]
	01	I2S ch.5 I2SMCLK 输出引脚使用 I2SMCLK5_0。
	10	I2S ch.5 I2SMCLK 输出引脚使用 I2SMCLK5_1。
	11	I2S ch.5 I2SMCLK 输出引脚使用 I2SMCLK5_2。

[bit17:16] MFSI2S MCLK5S：I2SMCLK5 输入选择位

选择 I2SMCLK5 输入。

位		描述
读取		读取寄存器值。
写入	00	不产生 I2S ch.5 I2SMCLK 的输入。[初始值]
	01	I2S ch.5 I2SMCLK 输入引脚使用 I2SMCLK5_0。
	10	I2S ch.5 I2SMCLK 输入引脚使用 I2SMCLK5_1。
	11	I2S ch.5 I2SMCLK 输入引脚使用 I2SMCLK5_2。

[bit15:12] 保留：保留位
这些位的读取值为 "0x0"。
写入这些位时，将其设置为 "0x0"。

[bit11:10] SDO4E：MFSI2S SDO4 输出选择位
选择 I2SDO4 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 I2S ch.4I2SDO 的输出。[初始值]
	01	I2S ch.4I2SDO 输出引脚使用 I2SDO4_0。
	10	I2S ch.4I2SDO 输出引脚使用 I2SDO4_1。
	11	I2S ch.4I2SDO 输出引脚使用 I2SDO4_2。

[bit9:8] SDI4S：MFSI2S I2SDI4 输入选择位
选择 I2SDI4 输入。

位		描述
读取		读取寄存器值。
写入	00	不产生 I2S ch.4I2SDI 的输入。[初始值]
	01	I2S ch.4 I2SDI 输入引脚使用 I2SDI4_0。
	10	I2S ch.4 I2SDI 输入引脚使用 I2SDI4_1。
	11	I2S ch.4 I2SDI 输入引脚使用 I2SDI4_2。

[bit7:6] MFSI2S WS4B： I2SWS4 输出选择位
选择 I2SWS4 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 I2S ch.4 I2SWS 的输出。[初始值]
	01	I2S ch.4 I2SWS 输出引脚使用 I2SWS4_0。
	10	I2S ch.4 I2SWS 输出引脚使用 I2SWS4_1。
	11	I2S ch.4 I2SWS 输出引脚使用 I2SWS4_2。

[bit5:4] MFSI2SSCK4B：I2SCK4 输出选择位
选择 I2SCK4 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 I2S ch.4 I2SCK 的输出。[初始值]
	01	I2S ch.4 I2SCK 输出引脚使用 I2SCK4_0。
	10	I2S ch.4 I2SCK 输出引脚使用 I2SCK4_1。
	11	I2S ch.4 I2SCK 输出引脚使用 I2SCK4_2。

[bit3:2] MFSI2SMCLK4E：I2SMCLK4 输出选择位
选择 I2SMCLK4 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 I2S ch.4 I2SMCLK 的输出。[初始值]
	01	I2S ch.4 I2SMCLK 输出引脚使用 I2SMCLK4_0。
	10	I2S ch.4 I2SMCLK 输出引脚使用 I2SMCLK4_1。
	11	I2S ch.4 I2SMCLK 输出引脚使用 I2SMCLK4_2。

[bit1:0] MFSI2S MCLK4S：I2SMCLK4 输入选择位

选择 I2SMCLK4 输入。

位		描述
读取		读取寄存器值。
写入	00	不产生 I2S ch.4 I2SMCLK 的输入。[初始值]
	01	I2S ch.4 I2SMCLK 输入引脚使用 I2SMCLK4_0。
	10	I2S ch.4 I2SMCLK 输入引脚使用 I2SMCLK4_1。
	11	I2S ch.4 I2SMCLK 输入引脚使用 I2SMCLK4_2。

4.32 扩展引脚功能设置寄存器 38 (EPFR38)

EPFR38 寄存器分配 MFS ch6 的 MFSI2S 功能。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	保留							
属性	-							
初始值	-							

位	23	22	21	20	19	18	17	16
字段	保留							
属性	-							
初始值	-							

位	15	14	13	12	11	10	9	8
字段	保留		保留		SDO6E		SDI6S	
属性	-		-		R/W		R/W	
初始值	-		-		00		00	

位	7	6	5	4	3	2	1	0
字段	WS6B		SCK6B		MCK6E		MCK6S	
属性	R/W		R/W		R/W		R/W	
初始值	00		00		00		00	

寄存器功能

[bit31:12] 保留：保留位

这些位的读取值为 "0"。

写入这些位时，将其设置为 "0"。

[bit11:10] SDO6E： MFSI2S SDO6 输出选择位

选择 I2SDO6 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 I2S ch.6I2SDO 的输出。[初始值]
	01	I2S ch.6I2SDO 输出引脚使用 I2SDO6_0。
	10	I2S ch.6I2SDO 输出引脚使用 I2SDO6_1。
	11	I2S ch.6I2SDO 输出引脚使用 I2SDO6_2。

[bit9:8] SDI6S：MFSI2S I2SDI6 输入选择位

选择 I2SDI6 输入。

位		描述
读取		读取寄存器值。
写入	00	不产生 I2S ch.6I2SDI 的输入。[初始值]
	01	I2S ch.6 I2SDI 输入引脚使用 I2SDI6_0。
	10	I2S ch.6 I2SDI 输入引脚使用 I2SDI6_1。
	11	I2S ch.6 I2SDI 输入引脚使用 I2SDI6_2。

[bit7:6] MFSI2S WS6B：I2SWS6 输出选择位

选择 I2SWS6 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 I2S ch.6 I2SWS 的输出。[初始值]
	01	I2S ch.6 I2SWS 输出引脚使用 I2SWS6_0。
	10	I2S ch.6 I2SWS 输出引脚使用 I2SWS6_1。
	11	I2S ch.6 I2SWS 输出引脚使用 I2SWS6_2。

[bit5:4] MFSI2SSCK6B：I2SCK6 输出选择位

选择 I2SCK6 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 I2S ch.6 I2SCK 的输出。[初始值]
	01	I2S ch.6 I2SCK 输出引脚使用 I2SCK6_0。
	10	I2S ch.6 I2SCK 输出引脚使用 I2SCK6_1。
	11	I2S ch.6 I2SCK 输出引脚使用 I2SCK6_2。

[bit3:2] MFSI2SMCLK6E：I2SMCLK6 输出选择位

选择 I2SMCLK6 输出。

位		描述
读取		读取寄存器值。
写入	00	不产生 I2S ch.6 I2SMCLK 的输出。[初始值]
	01	I2S ch.6 I2SMCLK 输出引脚使用 I2SMCLK6_0。
	10	I2S ch.6 I2SMCLK 输出引脚使用 I2SMCLK6_1。
	11	I2S ch.6 I2SMCLK 输出引脚使用 I2SMCLK6_2。

[bit1:0] MFSI2S MCLK6S：I2SMCLK6 输入选择位

选择 I2SMCLK6 输入。

位		描述
读取		读取寄存器值。
写入	00	不产生 I2S ch.6 I2SMCLK 的输入。[初始值]
	01	I2S ch.6 I2SMCLK 输入引脚使用 I2SMCLK6_0。
	10	I2S ch.6 I2SMCLK 输入引脚使用 I2SMCLK6_1。
	11	I2S ch.6 I2SMCLK 输入引脚使用 I2SMCLK6_2。

4.33 专用端口设置寄存器 (SPSR)

SPSR 寄存器将引脚设置为专用功能信号引脚。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	保留							
属性	-							
初始值	-							

位	23	22	21	20	19	18	17	16
字段	保留							
属性	-							
初始值	-							

位	15	14	13	12	11	10	9	8
字段	保留							
属性	-							
初始值	-							

位	7	6	5	4	3	2	1	0
字段	保留			USB0C	MAINXC		SUBXC	
属性	-			R/W	R/W		R/W	
初始值	-			0	01		01	

寄存器功能

[bit31:5] 保留：保留位

这些位的读取值为 "0"。

写入这些位时，将其设置为 "0"。

[bit4] USB0C： USB (ch.0) 引脚设置寄存器

此位将引脚设置成 USB 引脚。

bit4		描述
读取		读取寄存器值。
写入	0	UDM0 和 UDP0 两个引脚不用作 USB 引脚，而是用作数字输入/输出引脚。 [初始值]
	1	UDM0 和 UDP0 引脚用作 USB 引脚。 （I/O 单元将处于输入方向状态和输入切断状态。）

[bit3:2] MAINXC：主时钟（振荡）引脚设置寄存器

此位将引脚设置为主时钟（振荡）引脚。

bit3:2		描述
读取		读取寄存器值。
写入	00	X0 和 X1 两个引脚不用作主时钟（振荡）引脚，而是用作数字输入/输出引脚。
	01	X0 和 X1 引脚用作主时钟（振荡）引脚。[初始值] （I/O 单元将处于输入方向状态、输入切断状态和上拉断开状态。）
	10	禁止设置。
	11	X0 引脚用作外部时钟输入引脚。 X1 引脚用作数字输入/输出。

注意事项:

- *只将 "01" 写入 MAINXC 位时，主时钟不会起振。*
  启动振荡时，将 "01" 写入 MAINXC 位后，通过 "时钟" 一章所述系统时钟模式控制寄存器 (SCM_CTL) 的 MOSCE 位使能振荡。
- *使用外部时钟时，参见各产品《数据手册》"处理注意事项" 中的 "使用外部时钟"*

[bit1:0] SUBXC：副时钟（振荡）引脚设置寄存器

此位将引脚设置为 TYPE 1-M0+ 产品中的副时钟（振荡）引脚。

在 TYPE 2-M0+ 产品中，参见 “VBAT 域” 一章

在 TYPE 3-M0+ 产品中，参见 “低功耗模式” 一章

bit1:0		描述
读取		读取寄存器值。
写入	00	X0A 和 X1A 引脚不用作副时钟（振荡）引脚，而是用作数字输入/输出引脚。
	01	X0A 和 X1A 引脚用作副时钟（振荡）引脚。[初始值] （I/O 单元将处于输入方向状态、输入切断状态和上拉断开状态。）
	10	禁止设置。
	11	将 X0A 引脚用作外部时钟输入引脚。 将 X1A 引脚用作数字输入/输出。

注意事项:

- *只将 "01" 写入 SUBXC 位时，副时钟不会起振。*
  启动振荡时，将 "01" 写入 SUBXC 位后，通过 "时钟" 一章所述系统时钟模式控制寄存器 (SCM_CTL) 的 SOSCE 位使能振荡。
- *使用外部时钟时，参见各产品《数据手册》"处理注意事项" 中的 "使用外部时钟"。*

4.34 端口伪开漏设置寄存器 (PZRx)

PZRx 寄存器在输出高电平时设置 I/O 端口为 Hi-Z，并设置伪开漏控制。

PZR 寄存器配置表

位	31 16	15 0	初始值	属性
	保留	PZR0	0x0000	R/W
	保留	PZR1	0x0000	R/W
	保留	PZR2	0x0000	R/W
	保留	PZR3	0x0000	R/W
	保留	PZR4	0x0000	R/W
	保留	PZR5	0x0000	R/W
	保留	PZR6	0x0000	R/W
	保留	PZR7	0x0000	R/W
	保留	PZR8	0x0000	R/W
	保留	PZR9	0x0000	R/W
	保留	PZRA	0x0000	R/W
	保留	PZRB	0x0000	R/W
	保留	PZRC	0x0000	R/W
	保留	PZRD	0x0000	R/W
	保留	PZRE	0x0000	R/W
	保留	PZRF	0x0000	R/W

寄存器配置详情

位	31 16	15 0
字段	保留	PZRx

寄存器功能

[bit31:16] 保留：保留位

这些位的读取值为 "0x0000"。

写入这些位时，将其设置为 "0x0000"。

[bit15:0] PZRx：端口伪开漏设置寄存器 x

设置引脚伪开漏。

bit15:0		描述
读取		读取寄存器值。
写入	0	通过 GPIO 或外设宏输出数字高电平时，将引脚设置成高电平。
	1	通过 GPIO 或外设宏输出数字高电平时，将引脚设置成 Hi-Z。不管 PCR 如何设置，断开上拉电阻器。

注意事项:

- *PZRx 中的 "x" 是通配符，指 PZR0、PZR1、PZR2 等。*
- *只能通过一些专用引脚执行 PZR 寄存器功能。*
  只有《数据手册》"I/O 电路类型" 注释栏中 "使能 PZR 寄存器控制" 所述的引脚能控制开漏。

- *不是所有引脚中都有 PZR 寄存器。但是，如果不含 PZR 寄存器的引脚被用作 GPIO，也可以通过 DDR 设置控制伪开漏。*
  在这种情况下，在设置 PFR = 0（GPIO 设置）和 PDOR = 0 后，
  当设置 L 输出时：用作 DDR = 1（输出方向）。
  当设置 Hi-Z 输出时：用作 DDR = 0（输入方向）。

- *然而，在通过 GPIO 设置的开漏状态下，不能在 Hi-Z 应用超过 VCC 的电压。*

- *初始值仅仅是 TYPE 1-M0+ 产品的一个示例，有关不同产品类型，参见附录。*

4.35 LVDI 输入设置寄存器 (LVDIE)

如果产品配有 LVDI 引脚，LVDIE 寄存器设置使能输入 LVDI。

寄存器配置

位	31	30	29	28	27	26	25	24
字段	保留							
属性	-							
初始值	-							

位	23	22	21	20	19	18	17	16
字段	保留							
属性	-							
初始值	-							

位	15	14	13	12	11	10	9	8
字段	保留							
属性	-							
初始值	-							

位	7	6	5	4	3	2	1	0
字段	保留							LVDIE
属性	-							R/W
初始值	-							0

寄存器功能

[bit31:1] 保留：保留位

这些位的读取值为 "0x00000000"。

写入这些位时，将其设置为 "0"。

[bit0] LVDIE： LVD 输入使能设置寄存器

LVDIE 寄存器设置使能输入 LVDI。

bit15:0		描述
读取		读取寄存器值。
写入	0	通过 GPIO 或外设宏输出数字高电平时，将引脚设置成高电平。
	1	通过 GPIO 或外设宏输出数字高电平时，将引脚设置成 Hi-Z。不管 PCR 如何设置，断开上拉电阻器。

注意事项：

- *TYPE 1-M0+ 和 TYPE 3-M0+ 产品没有此寄存器。*
- *深度待机转换复位不会初始化此寄存器。*
- *有关 LVDIE 的详细说明，参见第 5-3 章。*

5. 使用注意事项

本节说明使用 I/O 端口的注意事项。

■ SPL=1 时，开/关上拉电阻。

SPL 是待机模式时将引脚转换为 Hi-Z 状态的信号。

- SPL=0 时　　正常操作
- SPL=1 时　　引脚为 Hi-Z，输入切断、上拉断开
  然而，SPL 位不能用于设置外部中断引脚、NMIX 引脚、SWD 引脚。

有关 SPL 位的详细说明，参见 "低功耗模式" 一章。

■ DTTIX 输入

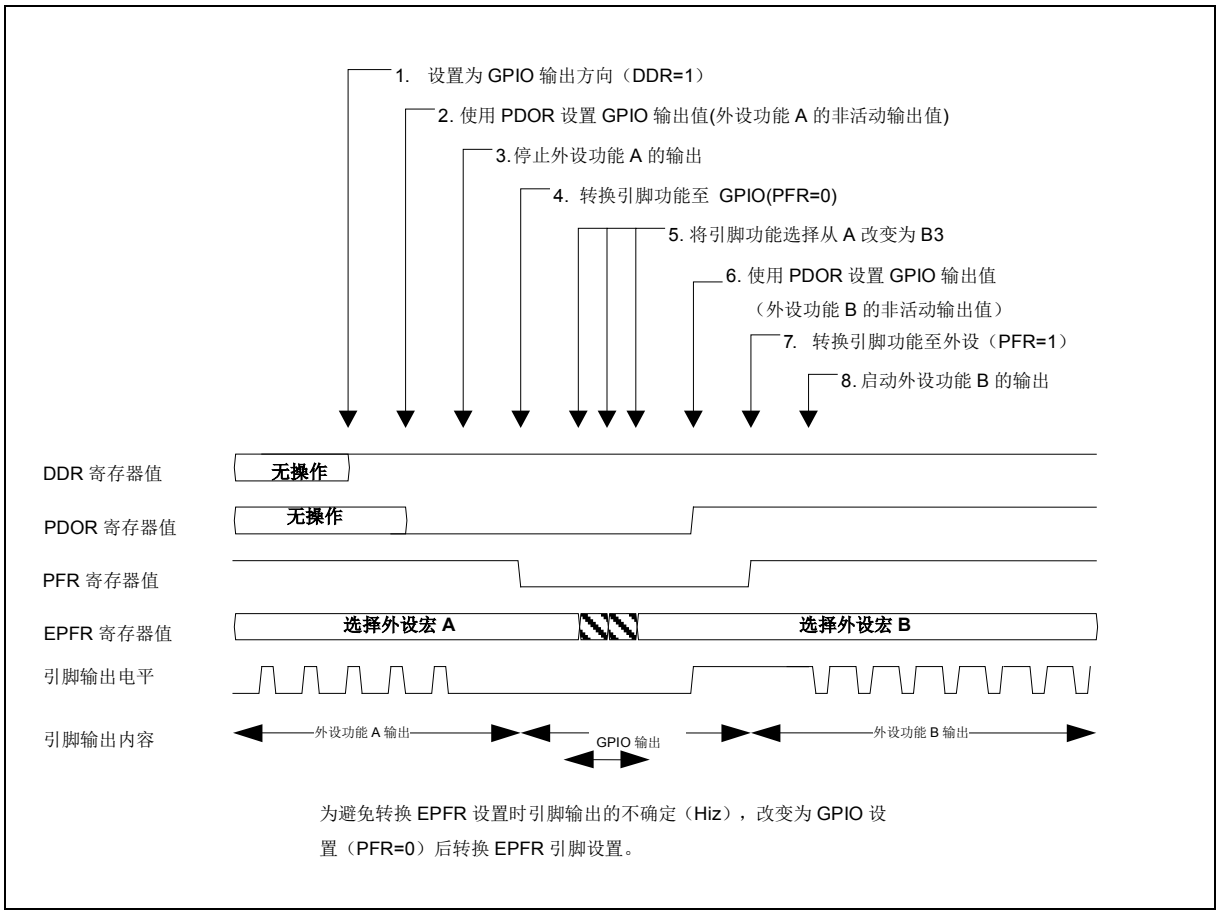
DTTI 输入是一种输入信号，将两用电机控制 PWM 输出 (RTO) 设置中的输出引脚转换为其他 GPIO 引脚，处理紧急情况下的电机停止需求。

使用本功能时，通过 EPFR 使能转换。

■ 引脚功能转换步骤

使用 EPFR 寄存器转换外设功能输出时，按照 Figure 5-1 所示转换示例相同的步骤进行转换设置，防止出现引脚不确定输出 (Hi-Z)。

Figure 5-1 引脚功能转换步骤



■ 保留位

此位读取值为 "0"，ADE 寄存器中此位读取值除外。写入时，始终写入 "0"。ADE 保留位的读取值为 "1"。写入时，始终写入 "1"。

■ 多功能串口引脚组

有多功能串口输入/输出时，将同一组端口设置成输入/输出。 "同一组端口" 指引脚名称上附加的重定位功能序号相同，就像 "xxx_0" 或 "yyy_1"。

Table 5-1 为设置示例。

Table 5-1 多功能串口设置示例

串口数据输出	串口时钟输入/输出	串口数据输入	有效端口
引脚 SOT1_0（端口 0）	引脚 SCK1_0（端口 0）	引脚 SIN1_0（端口 0）	端口 0
		引脚 SIN1_1（端口 1）	禁止设置。
	引脚 SCK1_1（端口 1）	引脚 SIN1_0（端口 0）	
		引脚 SIN1_1（端口 1）	
引脚 SOT1_1（端口 1）	引脚 SCK1_0（端口 0）	引脚 SIN1_0（端口 0）	
		引脚 SIN1_1（端口 1）	
	引脚 SCK1_1（端口 1）	引脚 SIN1（端口 0）	
		引脚 SIN1_1（端口 1）	端口 1

■ 外设功能输出

由于外设功能输出引脚是唯一由 EPFR 设置确定，单独的引脚不能分配外设功能输出。

（禁用示例）将多功能串口输出 SOT1_0 和 SOT1_1 分配给相同输出。

■ 引脚设置和操作模式

有关 SWD 设置，参见 "调试接口" 一章。

有关待机模式或复位进程中各引脚的状态，参见所用产品的《数据手册》。

■ 产品规格和外设功能引脚分配

引脚（GPO、外设 I/O 和专用 I/O）功能分配因产品不同而不同。参见《数据手册》中的引脚功能表，确定各产品的引脚功能。EPFR 寄存器设置时不能为产品中没有的引脚选择功能。

■ PE0 引脚用作 GPIO 时

需进行以下设置使用 PE0 引脚：

输入： 通过读取 PDIR 读取该值。

输出： 只有 L 输出，因为 PE0 引脚的 I/O 为 Nch 开漏引脚。

PFR=0（用作 GPIO。）

DDR=1（用作输出）

PDOR=0（输出数据为 "0"。）

SPL=0（在 STOP 模式下保持 GPIO 状态。）

■ 待机模式下的外部中断引脚设置

在 SPL=1 的设置下将模式转换为待机模式时，设置 PFR=1 并选择外设功能，以使能外部中断分配引脚进行唤醒。

如果用于外部中断的引脚设置保持为 PFR=0，将出现误操作。

第 10-2 章：高速 GPIO

本章说明高速 **GPIO** 端口。

1. 概述
2. 配置
3. 设置步骤示例
4. 寄存器
5. 位处理基址

代码：9AF_FastGPIO-C01.0

1. 概述

本节概述高速 GPIO 的操作。

1 个周期访问

- 高速 GPIO 指 GPIO 的 1 个周期访问。
- 高速 GPIO 可在 1 个 HCLK 时钟周期内读取输入电平并设置 CPU 的输出电平。
- GPIO 在初始状态下是选中的。需要通过寄存器设置选择高速 GPIO 才能使用高速 GPIO 功能。

引脚配置

所有 I/O 端口都具有普通 GPIO 和高速 GPIO。

位处理

高速 GPIO 支持通过专用电路进行位处理。
即使是在执行位处理时，也可以进行 1 个周期访问。

同时访问功能

镜像寄存器最多可同时访问 8 个端口，该寄存器将不同组的某些寄存器绑定在一起。这样使能各个产品的有效访问。

2. 配置

本节说明高速 GPIO 的配置、框图及操作。

高速 GPIO 的配置

Figure 2-1 所示为高速 GPIO 框图。
I/O 端口可通过 FPOER 在普通 GPIO 和高速 GPIO 之间切换。
设置 PCRx、DDRx，通过 GPIO 复用 ADE。

Figure 2-1 高速 GPIO 端口框图

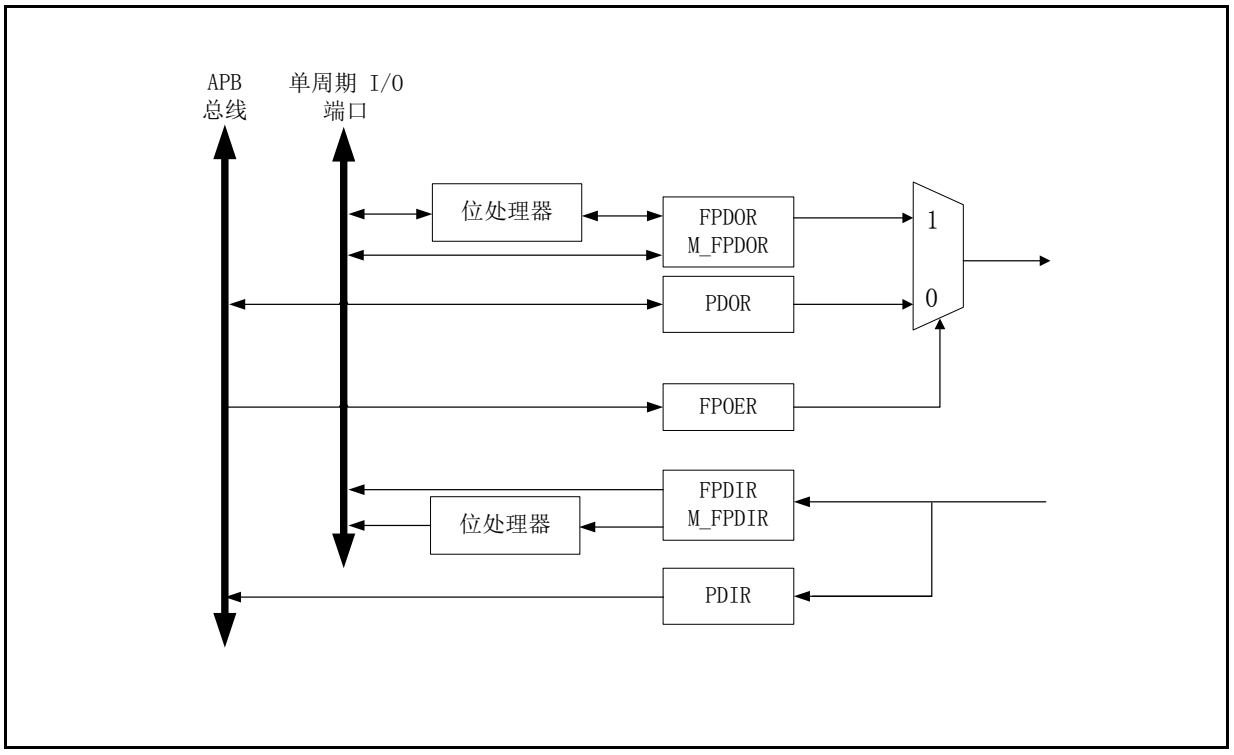


Table 2-1 寄存器功能

寄存器名称	功能
FPDIR	用于读取 I/O 端口电平状态的寄存器。此寄存器与单周期 I/O 端口连接，可通过 1 个周期访问。 - 当高速 GPIO 端口用作数字输入引脚时，本寄存器读取输入电平。 - 当高速 GPIO 端口用作数字输出引脚时，本寄存器读取输出电平。 - 当高速 GPIO 端口用作专用引脚时，本寄存器读取总为 "0"。
FPDOR	当 I/O 端口用做高速 GPIO 输出引脚时，此寄存器用来设置输出电平。此寄存器与单周期 I/O 端口连接，可通过 1 个周期访问。 - 设置为 "0" 时，输出低电平。 - 设置为 "1" 时，输出高电平。 注意事项： - 当引脚被选为 GPIO 功能的输入状态或外设功能输入/输出时，设置值无效。 - 此寄存器的设置会影响 M_FPDOR 相关位的值。
M_FPDIR	用于读取 FPDIR 电平状态的镜像寄存器。 可同时读取某些高速 GPIO 端口的输入值。 - 此寄存器的功能与 FPDIR 相同。
M_FPDOR	用于设置 FPDOR 输出电平的镜像寄存器。 可同时设置某些高速 GPIO 端口的输出值。 - 此寄存器的功能与 FPDOR 相同。 - 此寄存器的设置会影响 FPDOR 相关位的值。
FPOER	选择普通 GPIO 和高速 GPIO 端口输出的寄存器。 - 设置为 “0” 时选择普通 GPIO 端口输出。 - 设置为 “1” 时，选择高速 GPIO 端口输出。 - 此寄存器的设置对 FPDIR/M_FPDIR 无效。

高速 GPIO 访问

- 当 GPIO 配置为高速 GPIO 时，各引脚都可通过 FPDIR 读取并通过 FPDOR 设置输出电平。使用方法与 PDIR/PDOR 相同。
- FPDIR/PDIR 和 FPDOR/PDOR 的位之间是一一对应的关系。
- 例如：

  当 P00 设置为输入状态时，访问 FPDIR0 的 bit0 读取 P00 的输入值。

  当 P15 设置为高速 GPIO 和输出状态时，访问 FPDOR1 的 bit5 设置 P15 的输出值。

高速 GPIO 镜像访问 (TYPE 1-M0+, TYPE 2-M0+)

- 有些引脚可灵活地重新分配为专用组。
- 分配到镜像寄存器 (M_FPDIRx/M_FPDORx) 的一组端口可同时访问。
- Table 2-2 列出了 M_FPDIR0/M_FPDOR0 的配置。 P10-15 和 P22-23 可同时访问。所以不需要单独读取 FPDIR1 和 FPDIR2，也不需要单独写入 FPDOR1 和 FPDOR2。
- Table 2-3 列出了 M_FPDIR1/M_FPDOR1 的配置。 P3A-P3F 和 P46-47 可同时访问。所以不需要单独读取 FPDIR3 和 PFPDIR4，也不需要单独写入 FPDOR3 和 FPDOR4。

Table 2-2 M_FPDIR0/M_FPDOR0 配置（TYPE 1-M0+和 TYPE 2-M0+产品）

镜像端口名称	端口	对应 FPDIR 位	对应 FPDOR 位
M_FP00	P10	FPDIR1[0]	FPDOR1[0]
M_FP01	P11	FPDIR1[1]	FPDOR1[1]
M_FP02	P12	FPDIR1[2]	FPDOR1[2]
M_FP03	P13	FPDIR1[3]	FPDOR1[3]
M_FP04	P14	FPDIR1[4]	FPDOR1[4]
M_FP05	P15	FPDIR1[5]	FPDOR1[5]
M_FP06	P23	FPDIR2[3]	FPDOR2[3]
M_FP07	P22	FPDIR2[2]	FPDOR2[2]

Table 2-3M_FPDIR1/M_FPDOR1 配置(TYPE 1-M0+和 TYPE 2-M0+ 产品)

镜像端口名称	端口	对应 FPDIR 位	对应 FPDOR 位
M_FP10	P3A	FPDIR3[10]	FPDOR3[10]
M_FP11	P3B	FPDIR3[11]	FPDOR3[11]
M_FP12	P3C	FPDIR3[12]	FPDOR3[12]
M_FP13	P3D	FPDIR3[13]	FPDOR3[13]
M_FP14	P3E	FPDIR3[14]	FPDOR3[14]
M_FP15	P3F	FPDIR3[15]	FPDOR3[15]
M_FP16	P46	FPDIR4[6]	FPDOR4[6]
M_FP17	P47	FPDIR4[7]	FPDOR4[7]

注意事项：

- *写入 M_FPDORx 位会影响 FPDORx 相关位的值。例如，当 M_FPDOR1 的 bit5 设置为 "1" 时，FPDOR3 的 bit15 的值也变为 "1"。*

高速 GPIO 镜像访问 (TYPE 3-M0+ 产品)

- 有些引脚可灵活地重新分配为专用组。
- 分配到镜像寄存器 (M_FPDIR/M_FPDOR) 的一组端口可同时访问。
- Table 2-4 列出了 M_FPDIR0/M_FPDOR0 的配置。P10-15 和 P22-23 可同时访问。不需要单独读取 FPDIR1 和 FPDIR2。
- Table 2-5 列出了 M_FPDIR1/M_FPDOR1 的配置。 P05、P0F、P21、P46-47 和 P50-52 可同时访问。不需要单独读取 FPDIR0、FPDIR2、FPDIR4 和 PFDIR5。

Table 2-4 M_FPDIR0/M_FPDOR0 配置（TYPE 3-M0+产品）

镜像端口名称	端口	对应 FPDIR 位	对应 FPDOR 位
M_FP00	P10	FPDIR1[0]	FPDOR1[0]
M_FP01	P11	FPDIR1[1]	FPDOR1[1]
M_FP02	P12	FPDIR1[2]	FPDOR1[2]
M_FP03	P13	FPDIR1[3]	FPDOR1[3]
M_FP04	P14	FPDIR1[4]	FPDOR1[4]
M_FP05	P15	FPDIR1[5]	FPDOR1[5]
M_FP06	P23	FPDIR2[3]	FPDOR2[3]
M_FP07	P22	FPDIR2[2]	FPDOR2[2]

Table 2-5 M_FPDIR1/M_FPDOR1 配置 (TYPE 3-M0+产品)

镜像端口名称	端口	对应 FPDIR 位	对应 FPDOR 位
M_FP10	P21	FPDIR2[1]	FPDOR2[1]
M_FP11	P05	FPDIR0[5]	FPDOR0[5]
M_FP12	P0F	FPDIR0[15]	FPDOR0[15]
M_FP13	P50	FPDIR5[0]	FPDOR5[0]
M_FP14	P51	FPDIR5[1]	FPDOR5[1]
M_FP15	P52	FPDIR5[2]	FPDOR5[2]
M_FP16	P46	FPDIR4[6]	FPDOR4[6]
M_FP17	P47	FPDIR4[7]	FPDOR4[7]

注意事项：

- *写入 M_FPDORx 位会影响 FPDORx 相关位的值。例如，当 M_FPDOR1 的 bit5 设置为 "1" 时，FPDOR5 的 bit2 的值也变为 "1"。*

位处理

■ 概述

高速 GPIO 安装在单周期 I/O 端口。单周期 I/O 端口进行一个周期的访问数据。但是，单周期 I/O 端口不支持位带操作。

高速 GPIO 支持特殊操作的位处理。可单独设置/清除高速 GPIO 的各位。和位带一样，也有附加的位处理别名区。

位处理仅支持访问 FPDIR、FPDOR、M_FPDIR 和 M_FPDOR。

■ 使用方法

下面说明位处理地址

- 别名地址

  位处理的别名地址访问。公式如下：

别名地址=基址+偏移地址

- 基址

  基址由各个 FPDIR、FPDOR、M_FPDIR 和 M_FPDOR 决定。详见“5. 位处理基址”

- 偏移地址

  访问偏移地址设置访问位。

Table 2-6 位处理的偏移地址

访问位	偏移地址	访问位	偏移地址
bit8	0x0004+1	bit0	0x0004
bit9	0x0008+1	bit1	0x0008
bit10	0x0010+1	bit2	0x0010
bit11	0x0020+1	bit3	0x0020
bit12	0x0040+1	bit4	0x0040
bit13	0x0080+1	bit5	0x0080
bit14	0x0100+1	bit6	0x0100
bit15	0x0200+1	bit7	0x0200
禁止设置	其它	禁止设置	其它

FPDIR 读取值只反映 "访问位"，其他位为 "0"。例如，当 FPDIR8 为 "0x07" 时，"FPDIR8+基址+0x0010" 读取值为 "0x04"。

写入 FPDOR 只对 "访问位" 有效，其他位忽略。例如，当 FPDOR1 为 "0x00" 且写入 "0xFF" 至 "FPDOR1+基址+0x0008" 时，FPDOR1 会变为 "0x02"。

■ 别名地址计算示例

– 设置 FPDOR0 的 bit3 时

访问寄存器	访问位	访问量	基址	偏移地址	别名地址
FPDOR0	bit3	字节	0xF801_4000	0x0020	0xF801_4020

– 设置 FPDOR0 的 bit13 时

访问寄存器	访问位	访问量	基址	偏移地址	别名地址
FPDOR0	bit13	字节	0xF801_4000	0x0080+1	0xF801_4081

半字访问时，高位字节和低位字节都被屏蔽。

– 设置 FPDOR0 的 bit13 和 bit5 时

访问寄存器	访问位	访问量	基址	偏移地址	别名地址
FPDOR0	bit13, bit5	半字	0xF801_4000	0x0080	0xF801_4080

字访问与半字访问相同。

3. 设置步骤示例

本节以示例说明高速 GPIO 的设置步骤。

设置高速 GPIO

1. 当高速 GPIO 用作输出时，通过设置 FPOER 寄存器选择高速 GPIO。
2. 将 I/O 端口寄存器设置为普通 GPIO。

Figure 3-1 所示为输出设置步骤的示例。
Figure 3-2 所示为输入设置步骤的示例。

Figure 3-1 高速 GPIO 输出设置步骤示例

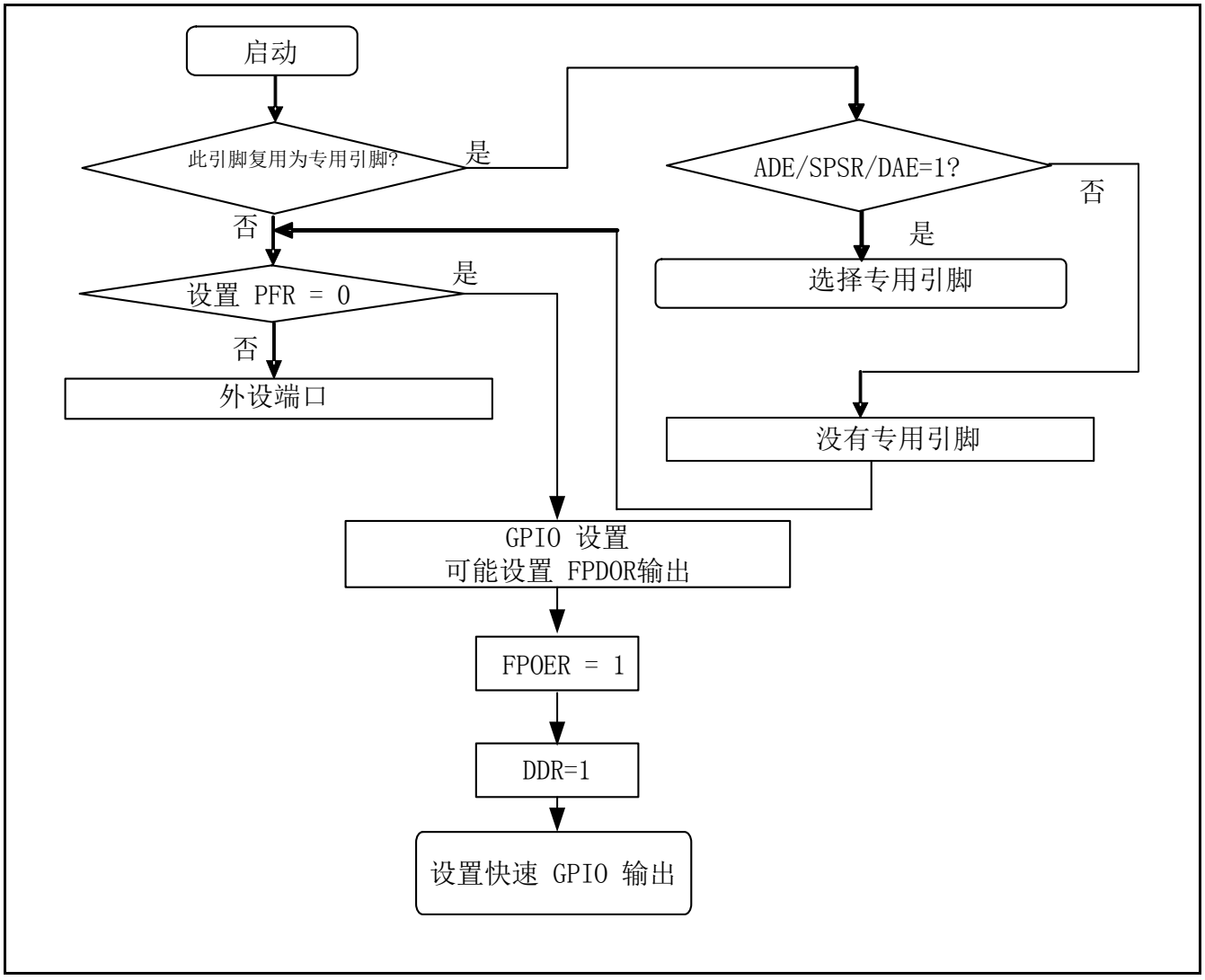
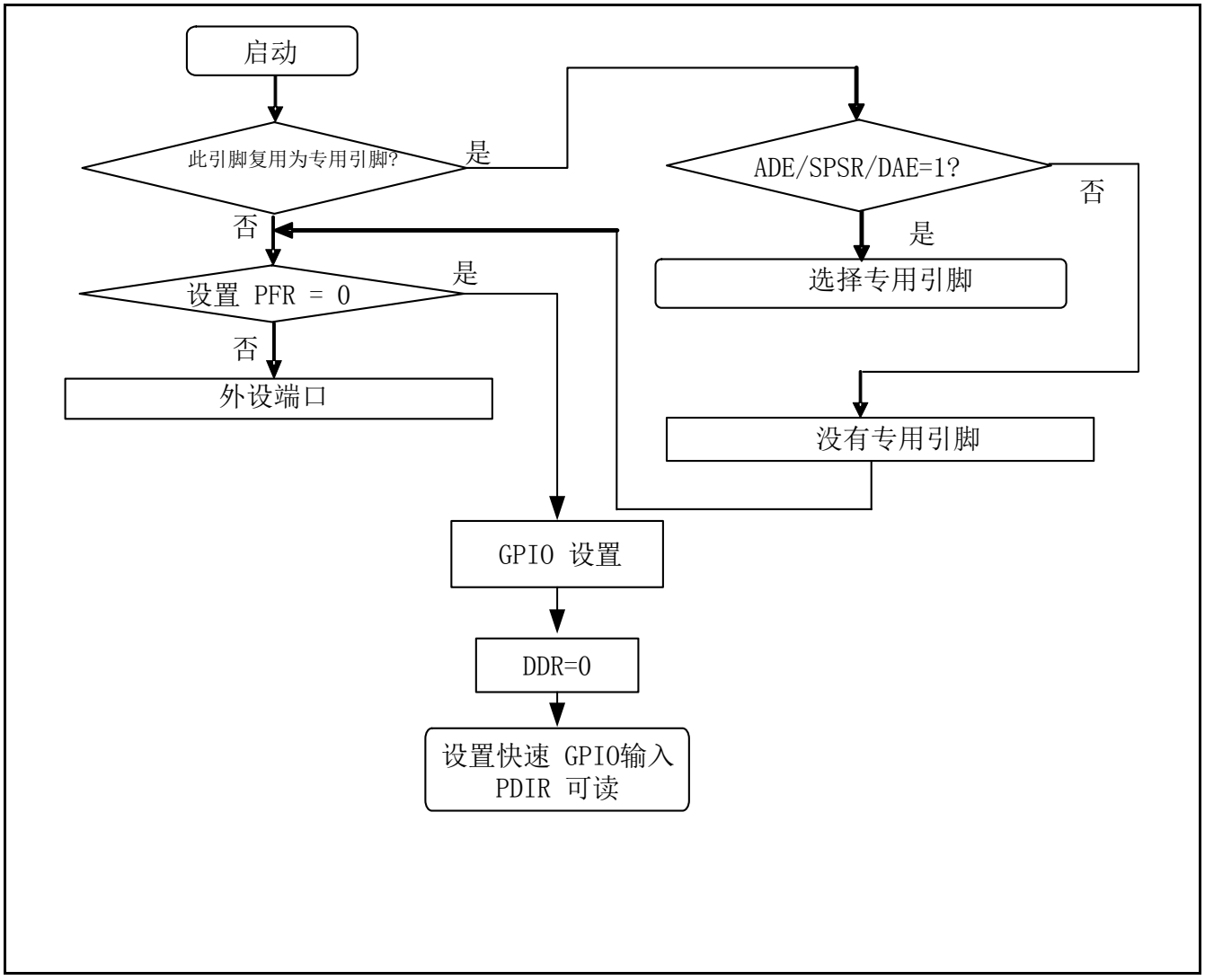


Figure 3-2 设置高速 GPIO 输入设置步骤示例



4. 寄存器

本节描述高速 GPIO 端口的寄存器。

高速 GPIO 端口寄存器列表

Table 4-1 高速 GPIO 端口寄存器列表

缩写	寄存器名称	参考章节
FPDIR0	高速 GPIO 输入数据寄存器 0	4.1
FPDIR1	高速 GPIO 输入数据寄存器 1	
FPDIR2	高速 GPIO 输入数据寄存器 2	
FPDIR3	高速 GPIO 输入数据寄存器 3	
FPDIR4	高速 GPIO 输入数据寄存器 4	
FPDIR5	高速 GPIO 输入数据寄存器 5	
FPDIR6	高速 GPIO 输入数据寄存器 6	
FPDIR7	高速 GPIO 输入数据寄存器 7	
FPDIR8	高速 GPIO 输入数据寄存器 8	
FPDIR9	高速 GPIO 输入数据寄存器 9	
FPDIRA	高速 GPIO 输入数据寄存器 A	
FPDIRB	高速 GPIO 输入数据寄存器 B	
FPDIRC	高速 GPIO 输入数据寄存器 C	
FPDIRD	高速 GPIO 输入数据寄存器 D	
FPDIRE	高速 GPIO 输入数据寄存器 E	
FPDIRF	高速 GPIO 输入数据寄存器 F	
FPDOR0	高速 GPIO 输出数据寄存器 0	4.2
FPDOR1	高速 GPIO 输出数据寄存器 1	
FPDOR2	高速 GPIO 输出数据寄存器 2	
FPDOR3	高速 GPIO 输出数据寄存器 3	
FPDOR4	高速 GPIO 输出数据寄存器 4	
FPDOR5	高速 GPIO 输出数据寄存器 5	
FPDOR6	高速 GPIO 输出数据寄存器 6	
FPDOR7	高速 GPIO 输出数据寄存器 7	
FPDOR8	高速 GPIO 输出数据寄存器 8	
FPDOR9	高速 GPIO 输出数据寄存器 9	
FPDORA	高速 GPIO 输出数据寄存器 A	
FPDORB	高速 GPIO 输出数据寄存器 B	
FPDORC	高速 GPIO 输出数据寄存器 C	
FPDORD	高速 GPIO 输出数据寄存器 D	
FPDORE	高速 GPIO 输出数据寄存器 E	
FPDORF	高速 GPIO 输出数据寄存器 F	

缩写	寄存器名称	参考章节
M_FPDIR0	镜像高速 GPIO 输入数据寄存器 0	4.3
M_FPDIR1	镜像高速 GPIO 输入数据寄存器 1	
M_FPDIR2	镜像高速 GPIO 输入数据寄存器 2	
M_FPDIR3	镜像高速 GPIO 输入数据寄存器 3	
M_FPDIR4	镜像高速 GPIO 输入数据寄存器 4	
M_FPDIR5	镜像高速 GPIO 输入数据寄存器 5	
M_FPDIR6	镜像高速 GPIO 输入数据寄存器 6	
M_FPDIR7	镜像高速 GPIO 输入数据寄存器 7	
M_FPDIR8	镜像高速 GPIO 输入数据寄存器 8	
M_FPDIR9	镜像高速 GPIO 输入数据寄存器 9	
M_FPDIRA	镜像高速 GPIO 输入数据寄存器 A	
M_FPDIRB	镜像高速 GPIO 输入数据寄存器 B	
M_FPDIRC	镜像高速 GPIO 输入数据寄存器 C	
M_FPDIRD	镜像高速 GPIO 输入数据寄存器 D	
M_FPDIRE	镜像高速 GPIO 输入数据寄存器 E	
M_FPDIRF	镜像高速 GPIO 输入数据寄存器 F	
M_FPDOR0	镜像高速 GPIO 输出数据寄存器 0	4.4
M_FPDOR1	镜像高速 GPIO 输出数据寄存器 1	
M_FPDOR2	镜像高速 GPIO 输出数据寄存器 2	
M_FPDOR3	镜像高速 GPIO 输出数据寄存器 3	
M_FPDOR4	镜像高速 GPIO 输出数据寄存器 4	
M_FPDOR5	镜像高速 GPIO 输出数据寄存器 5	
M_FPDOR6	镜像高速 GPIO 输出数据寄存器 6	
M_FPDOR7	镜像高速 GPIO 输出数据寄存器 7	
M_FPDOR8	镜像高速 GPIO 输出数据寄存器 8	
M_FPDOR9	镜像高速 GPIO 输出数据寄存器 9	
M_FPDORA	镜像高速 GPIO 输出数据寄存器 A	
M_FPDORB	镜像高速 GPIO 输出数据寄存器 B	
M_FPDORC	镜像高速 GPIO 输出数据寄存器 C	
M_FPDORD	镜像高速 GPIO 输出数据寄存器 D	
M_FPDORE	镜像高速 GPIO 输出数据寄存器 E	
M_FPDORF	镜像高速 GPIO 输出数据寄存器 F	

缩写	寄存器名称	参考章节
FPOER0	高速 GPIO 输出使能寄存器 0	4.5
FPOER1	高速 GPIO 输出使能寄存器 1	
FPOER2	高速 GPIO 输出使能寄存器 2	
FPOER3	高速 GPIO 输出使能寄存器 3	
FPOER4	高速 GPIO 输出使能寄存器 4	
FPOER5	高速 GPIO 输出使能寄存器 5	
FPOER6	高速 GPIO 输出使能寄存器 6	
FPOER7	高速 GPIO 输出使能寄存器 7	
FPOER8	高速 GPIO 输出使能寄存器 8	
FPOER9	高速 GPIO 输出使能寄存器 9	
FPOERA	高速 GPIO 输出使能寄存器 A	
FPOERB	高速 GPIO 输出使能寄存器 B	
FPOERC	高速 GPIO 输出使能寄存器 C	
FPOERD	高速 GPIO 输出使能寄存器 D	
FPOERE	高速 GPIO 输出使能寄存器 E	
FPOERF	高速 GPIO 输出使能寄存器 F	

4.1 高速 GPIO 输入数据寄存器(FPDIRx)

高速 GPIO 输入数据寄存器(FPDIRx)用于指示引脚的输入数据。

FPDIRx 寄存器配置表

位 31 ~ 16	15 ~ 0	初始值	属性	对应端口
保留	FPDIR0	0xXXXX	R	P0F ~ P00
保留	FPDIR1	0xXXXX	R	P1F ~ P10
保留	FPDIR2	0xXXXX	R	P2F ~ P20
保留	FPDIR3	0xXXXX	R	P3F ~ P30
保留	FPDIR4	0xXXXX	R	P4F ~ P40
保留	FPDIR5	0xXXXX	R	P5F ~ P50
保留	FPDIR6	0xXXXX	R	P6F ~ P60
保留	FPDIR7	0xXXXX	R	P7F ~ P70
保留	FPDIR8	0xXXXX	R	P8F ~ P80
保留	FPDIR9	0xXXXX	R	P9F ~ P90
保留	FPDIRA	0xXXXX	R	PAF ~ PA0
保留	FPDIRB	0xXXXX	R	PBF ~ PB0
保留	FPDIRC	0xXXXX	R	PCF ~ PC0
保留	FPDIRD	0xXXXX	R	PDF ~ PD0
保留	FPDIRE	0xXXXX	R	PEF ~ PE0
保留	FPDIRF	0xXXXX	R	PFF ~ PF0

寄存器详细配置

位	31 ~ 16	15 ~ 0
字段	保留	FPDIRx

寄存器功能

[bit31:16] 保留：保留位

这些位的读取值为 "0x0000"。

写入这些位时，将其设置为 "0x0000"。

[bit15:0] FPDIRx：高速 GPIO 输入数据寄存器 x

读取高速 GPIO 的输入数据。

bit15:0		描述
读取	0	不管引脚功能如何设置 (PFR/EPFR/DDR/FPDOR/M_FPDOR)，此位表明引脚处于 "L" 电平输入或 "L" 电平输出状态。通过 ADE/SPSR 选择专用引脚时，由于输入被切断，读取值始终为 0。
	1	不管引脚功能如何设置 (PFR/EPFR/DDR/FPDOR/M_FPDOR)，此位表明引脚处于 "H" 电平输入或 "H" 电平输出状态。
写入		写入无效。

注意事项：

- *FPDIRx 中的 "x" 是通配符。 FPDIRx 表示 FPDIR0、FPDIR1、FPDIR2 等。*
- *Px0 和 PxF 中的 "x" 是通配符。 Px0 指 P00、P10、P20 等；PxF 指 P0F、P1F、P2F 等。*
- *一个寄存器可读取 PxF 至 Px0 共 16 个端口的输入数据。*
- *寄存器各位分别指示各引脚的状态。位分配和引脚序列一一对应。例如，FPDIR0 的第 15 位指示 P0F，FPDIR0 的第 14 位指示 P0E，FPDIR0 的第 0 位指示 P00。*
- *产品没有配置的引脚对应位的值总是读为 “0”。*

若为 TYPE 2-M0+ 产品

- *深度待机转换复位时不初始化 FPDIRx 寄存器。*

若为 TYPE 3-M0+ 产品

- *当 I/O 端口被深度待机转换复位锁存后，FPDIRx 寄存器被初始化。*

4.2 高速 GPIO 输出数据寄存器 x (FPDORx)

高速 GPIO 输出数据寄存器 x (FPDORx) 用于设置引脚的输出数据。

FPDORx 寄存器配置表

位 31 ~ 16	位 15 ~ 0	初始值	属性	对应端口
保留	FPDOR0	0x0000	R/W	P0F ~ P00
保留	FPDOR1	0x0000	R/W	P1F ~ P10
保留	FPDOR2	0x0000	R/W	P2F ~ P20
保留	FPDOR3	0x0000	R/W	P3F ~ P30
保留	FPDOR4	0x0000	R/W	P4F ~ P40
保留	FPDOR5	0x0000	R/W	P5F ~ P50
保留	FPDOR6	0x0000	R/W	P6F ~ P60
保留	FPDOR7	0x0000	R/W	P7F ~ P70
保留	FPDOR8	0x0000	R/W	P8F ~ P80
保留	FPDOR9	0x0000	R/W	P9F ~ P90
保留	FPDORA	0x0000	R/W	PAF ~ PA0
保留	FPDORB	0x0000	R/W	PBF ~ PB0
保留	FPDORC	0x0000	R/W	PCF ~ PC0
保留	FPDORD	0x0000	R/W	PDF ~ PD0
保留	FPDORE	0x0000	R/W	PEF ~ PE0
保留	FPDORF	0x0000	R/W	PFF ~ PF0

寄存器详细配置

位	31 ~ 16	15 ~ 0
字段	保留	FPDORx

寄存器功能

[bit31:16] 保留：保留位

这些位的读取值为 "0x0000"。

写入这些位时，将其设置为 "0x0000"。

[bit15:0] FPDORx：高速 GPIO 输出数据寄存器 x

设置高速 GPIO 的输出数据。

bit15:0		描述
读取		读取寄存器值。
写入	0	向高速 GPIO 输出 "L" 电平。 当引脚被选为 I/O 输入或普通 GPIO 或外设功能输入/输出时，设置值无效。
	1	向高速 GPIO 输出 "H" 电平。 当引脚被选为 I/O 输入或普通 GPIO 或外设功能输入/输出时，设置值无效。

注意事项：

- *FPDORx 中的 "x" 是通配符。 FPDORx 表示 FPDOR0、FPDOR1、FPDOR2 等。*
- *Px0 和 PxF 中的 "x" 是通配符。 Px0 指 P00、P10、P20 等；PxF 指 P0F、P1F、P2F 等。*
- *一个寄存器可设置 PxF 至 Px0 共 16 个端口的输出数据。*
- *寄存器各位分别设置各引脚。位分配和引脚序列一一对应。例如，FPDOR0 的第 15 位设置 P0F，FPDOR0 的第 14 位设置 P0E，FPDOR0 的第 0 位设置 P00。*
- *产品没有配置的引脚，相应位写入值无效且读取值未定义。*

若为 TYPE 2-M0+ 产品

- *深度待机转换复位时不初始化 FPDORx 寄存器。*

若为 TYPE 3-M0+ 产品

- *当 I/O 端口被深度待机转换复位锁存后，FPDORx 寄存器被初始化。*

4.3 镜像高速 GPIO 输入数据寄存器(M_FPDIRx)

镜像高速 GPIO 输入数据寄存器(M_FPDIRx)用于指示引脚的输入数据。

M_FPDIRx 寄存器配置表

位 31 ~ 8	7 ~ 0	初始值	属性	对应端口
保留	M_FPDIR0	0xXX	R	M_FP07 ~ M_FP00
保留	M_FPDIR1	0xXX	R	M_FP17 ~ M_FP10
保留	M_FPDIR2	0xXX	R	M_FP27 ~ M_FP20
保留	M_FPDIR3	0xXX	R	M_FP37 ~ M_FP30
保留	M_FPDIR4	0xXX	R	M_FP47 ~ M_FP40
保留	M_FPDIR5	0xXX	R	M_FP57 ~ M_FP50
保留	M_FPDIR6	0xXX	R	M_FP67 ~ M_FP60
保留	M_FPDIR7	0xXX	R	M_FP77 ~ M_FP70
保留	M_FPDIR8	0xXX	R	M_FP87 ~ M_FP80
保留	M_FPDIR9	0xXX	R	M_FP97 ~ M_FP90
保留	M_FPDIRA	0xXX	R	M_FPA7 ~ M_FPA0
保留	M_FPDIRB	0xXX	R	M_FPB7 ~ M_FPB0
保留	M_FPDIRC	0xXX	R	M_FPC7 ~ M_FPC0
保留	M_FPDIRD	0xXX	R	M_FPD7 ~ M_FPD0
保留	M_FPDIRE	0xXX	R	M_FPE7 ~ M_FPE0
保留	M_FPDIRF	0xXX	R	M_FPF7 ~ M_FPF0

寄存器详细配置

位	31 ~ 8	7 ~ 0
字段	保留	M_FPDIRx

寄存器功能

[bit31:8] 保留：保留位

这些位的读取值为 "0x0000"。写入这些位时，将其设置为 "0x0000"。

[bit7:0] M_FPDIRx：镜像高速 GPIO 输入数据寄存器 x

读取高速 GPIO 的输入数据。

bit7:0		Description
读取	0	不管引脚功能如何设置 (PFR/EPFR/DDR/FPDOR/M_FPDOR)，此位表明引脚处于 "L" 电平输入或 "L" 电平输出状态。通过 ADE/SPSR 选择专用引脚时，由于输入被切断，读取值始终为 0。
	1	不管引脚功能如何设置 (PFR/EPFR/DDR/FPDOR/M_FPDOR)，此位表明引脚处于 "H" 电平输入或 "H" 电平输出状态。
写入		写入无效。

注意事项：

- *M_FPDIRx 中的 "x" 是通配符。 M_FPDIRx 表示 M_FPDIR0、M_FPDIR1 等。*
- *产品没有配置的引脚对应位的值总是读为 "0"。*

若为 TYPE 2-M0+ 产品

- *深度待机转换复位时不初始化 M_FPDIRx 寄存器。*

若为 TYPE 3-M0+ 产品

- *当 I/O 端口被深度待机转换复位锁存后，N_FPDIRx 寄存器被初始化。*

4.4 镜像高速 GPIO 输出数据寄存器(M_FPDORx)

镜像高速 GPIO 输出数据寄存器(M_FPDORx)用于设置引脚的输出数据。

M_FPDORx 寄存器配置表

位 31 ~ 8	7 ~ 0	初始值	属性	对应端口
保留	M_FPDOR0	0x00	R/W	M_FP07 ~ M_FP00
保留	M_FPDOR1	0x00	R/W	M_FP17 ~ M_FP10
保留	M_FPDOR2	0x00	R/W	M_FP27 ~ M_FP20
保留	M_FPDOR3	0x00	R/W	M_FP37 ~ M_FP30
保留	M_FPDOR4	0x00	R/W	M_FP47 ~ M_FP40
保留	M_FPDOR5	0x00	R/W	M_FP57 ~ M_FP50
保留	M_FPDOR6	0x00	R/W	M_FP67 ~ M_FP60
保留	M_FPDOR7	0x00	R/W	M_FP77 ~ M_FP70
保留	M_FPDOR8	0x00	R/W	M_FP87 ~ M_FP80
保留	M_FPDOR9	0x00	R/W	M_FP97 ~ M_FP90
保留	M_FPDORA	0x00	R/W	M_FPA7 ~ M_FPA0
保留	M_FPDORB	0x00	R/W	M_FPB7 ~ M_FPB0
保留	M_FPDORC	0x00	R/W	M_FPC7 ~ M_FPC0
保留	M_FPDORD	0x00	R/W	M_FPD7 ~ M_FPD0
保留	M_FPDORE	0x00	R/W	M_FPE7 ~ M_FPE0
保留	M_FPDORF	0x00	R/W	M_FPF7 ~ M_FPF0

寄存器详细配置

位	31 ~ 8	7 ~ 0
字段	保留	M_FPDORx

寄存器功能

[bit31:8] 保留：保留位

这些位的读取值为 "0x0000"。写入这些位时，将其设置为 "0x0000"。

[bit7:0] M_FPDORx：镜像高速 GPIO 输出数据寄存器 x

设置高速 GPIO 的输出数据。

bit7:0		Description
读取		读取寄存器值。
写入	0	向高速 GPIO 输出 "L" 电平。 当引脚被选为 I/O 输入或普通 GPIO 或外设功能输入/输出时，设置值无效。
	1	向高速 GPIO 输出 "H" 电平。 当引脚被选为 I/O 输入或普通 GPIO 或外设功能输入/输出时，设置值无效。

注意事项：

- *M_FPDORx 中的 "x" 是通配符。 M_FPDORx 表示 M_FPDOR0、M_FPDOR1 等。*
- *产品没有配置的引脚，相应位写入值无效且读取值未定义。*

若为 TYPE 2-M0+ 产品

- *深度待机转换复位时不初始化 M_FPDORx 寄存器。*

若为 TYPE 3-M0+ 产品

- *当 I/O 端口被深度待机转换复位锁存后，N_FPDORx 寄存器被初始化。*

4.5 高速 GPIO 输出使能寄存器(FPOERx)

高速 GPIO 输出使能寄存器(FPOERx) 用于选择普通 GPIO/高速 GPIO 的输出。

FPOER 寄存器配置表

位 31 ~ 16	15 ~ 0	初始值	属性	对应端口
保留	FPOER0	0x0000	(*1)	P0F ~ P00
保留	FPOER1	0x0000	(*1)	P1F ~ P10
保留	FPOER2	0x0000	(*1)	P2F ~ P20
保留	FPOER3	0x0000	(*1)	P3F ~ P30
保留	FPOER4	0x0000	(*1)	P4F ~ P40
保留	FPOER5	0x0000	(*1)	P5F ~ P50
保留	FPOER6	0x0000	(*1)	P6F ~ P60
保留	FPOER7	0x0000	(*1)	P7F ~ P70
保留	FPOER8	0x0000	(*1)	P8F ~ P80
保留	FPOER9	0x0000	(*1)	P9F ~ P90
保留	FPOERA	0x0000	(*1)	PAF ~ PA0
保留	FPOERB	0x0000	(*1)	PBF ~ PB0
保留	FPOERC	0x0000	(*1)	PCF ~ PC0
保留	FPOERD	0x0000	(*1)	PDF ~ PD0
保留	FPOERE	0x0000	(*1)	PEF ~ PE0
保留	FPOERF	0x0000	(*1)	PFF ~ PF0

*1 在 TYPE 3-M0+ 产品中，属性为 R/W。在其它类型的产品中，属性为 W。

寄存器详细配置

位	31 ~ 16	15 ~ 0
字段	保留	FPOERx

寄存器功能

[bit31:16] Reserved: Reserved bits

这些位的读取值为 "0x0000"。

写入这些位时，将其设置为 "0x0000"。

[bit15:0] FPOERx：高速 GPIO 输出使能寄存器 x

I/O 端口输设置为普通 GPIO 或高速 GPIO。

bit15:0		描述
读取		读取值未定义。
写入	0	该引脚选择 PDOR 输出。 当引脚选为 I/O 输入或外设功能输入/输出时，设置值无效。
	1	该引脚选择 FPDOR 输出。 当引脚被选为 I/O 输入或外设功能输入/输出时，设置值无效。

注意事项：

- *FPOERx 寄存器安装在 APB 总线上。*
- *FPOERx 中的 "x" 是通配符。 FPOERx 表示 FPOER0、FPOER1、FPOER2 等。*
- *Px0 和 PxF 中的 "x" 是通配符。 Px0 指 P00、P10、P20 等； PxF 指 P0F、P1F、P2F 等。*
- *产品没有配置的引脚，相应位写入值无效且读取值未定义。*
  不管 FPOERx 的值为何，都可通过 FPDIRx 读取引脚的值。

若为 TYPE 1-M0+ 和 TYPE 2-M0+ 产品

- *这些寄存器为只写属性，不支持位带访问。*
- *深度待机转换复位时不初始化 FPOERx 寄存器。*

若为 TYPE 3-M0+ 产品

- *这些寄存器可读/写，支持位带访问。*
- *当 I/O 端口被深度待机转换复位锁存后，FPOERx 寄存器被初始化。*

5. 位处理基址

本节说明 FPDIRx、FPDORx、M_FPDIRx 和 M_FPDORx 的位处理基址。

基址表

Table 5-1 列出了位处理基址。

Table 5-1 别名区列表

名称	地址	名称	地址
FPDIR0 基址	0xF801_0000	FPDOR0 基址	0xF801_4000
FPDIR1 基址	0xF801_0400	FPDOR1 基址	0xF801_4400
FPDIR2 基址	0xF801_0800	FPDOR2 基址	0xF801_4800
FPDIR3 基址	0xF801_0C00	FPDOR3 基址	0xF801_4C00
FPDIR4 基址	0xF801_1000	FPDOR4 基址	0xF801_5000
FPDIR5 基址	0xF801_1400	FPDOR5 基址	0xF801_5400
FPDIR6 基址	0xF801_1800	FPDOR6 基址	0xF801_5800
FPDIR7 基址	0xF801_1C00	FPDOR7 基址	0xF801_5C00
FPDIR8 基址	0xF801_2000	FPDOR8 基址	0xF801_6000
FPDIR9 基址	0xF801_2400	FPDOR9 基址	0xF801_6400
FPDIRA 基址	0xF801_2800	FPDORA 基址	0xF801_6800
FPDIRB 基址	0xF801_2C00	FPDORB 基址	0xF801_6C00
FPDIRC 基址	0xF801_3000	FPDORC 基址	0xF801_7000
FPDIRD 基址	0xF801_3400	FPDORD 基址	0xF801_7400
FPDIRE 基址	0xF801_3800	FPDORE 基址	0xF801_7800
FPDIRF 基址	0xF801_3C00	FPDORF 基址	0xF801_7C00
M_FPDIR0 基址	0xF801_8000	M_FPDOR0 基址	0xF801_C000
M_FPDIR1 基址	0xF801_8400	M_FPDOR1 基址	0xF801_C400
M_FPDIR2 基址	0xF801_8800	M_FPDOR2 基址	0xF801_C800
M_FPDIR3 基址	0xF801_8C00	M_FPDOR3 基址	0xF801_CC00
M_FPDIR4 基址	0xF801_9000	M_FPDOR4 基址	0xF801_D000
M_FPDIR5 基址	0xF801_9400	M_FPDOR5 基址	0xF801_D400
M_FPDIR6 基址	0xF801_9800	M_FPDOR6 基址	0xF801_D800
M_FPDIR7 基址	0xF801_9C00	M_FPDOR7 基址	0xF801_DC00
M_FPDIR8 基址	0xF801_A000	M_FPDOR8 基址	0xF801_E000
M_FPDIR9 基址	0xF801_A400	M_FPDOR9 基址	0xF801_E400
M_FPDIRA 基址	0xF801_A800	M_FPDORA 基址	0xF801_E800
M_FPDIRB 基址	0xF801_AC00	M_FPDORB 基址	0xF801_EC00
M_FPDIRC 基址	0xF801_B000	M_FPDORC 基址	0xF801_F000
M_FPDIRD 基址	0xF801_B400	M_FPDORD 基址	0xF801_F400
M_FPDIRE 基址	0xF801_B800	M_FPDORE 基址	0xF801_F800
M_FPDIRF 基址	0xF801_BC00	M_FPDORF 基址	0xF801_FC00

第 11 章：CRC (循环冗余校验)

本章说明 **CRC** 功能。

1. **CRC 概述**
2. **CRC 操作**
3. **CRC 寄存器**

代码：FS15-C02.3

1. CRC 概述

CRC（循环冗余校验）是一个错误检测系统。假设输入数据串为高阶多项式，CRC 代码为输入数据串除以预定义生成多项式的余数。数据串在发送时通常加上 CRC 代码的后缀，收到的数据则除以生成多项式（如前所述）。如果接收到的数据可除尽，则判定数据接收正确。

CRC 功能

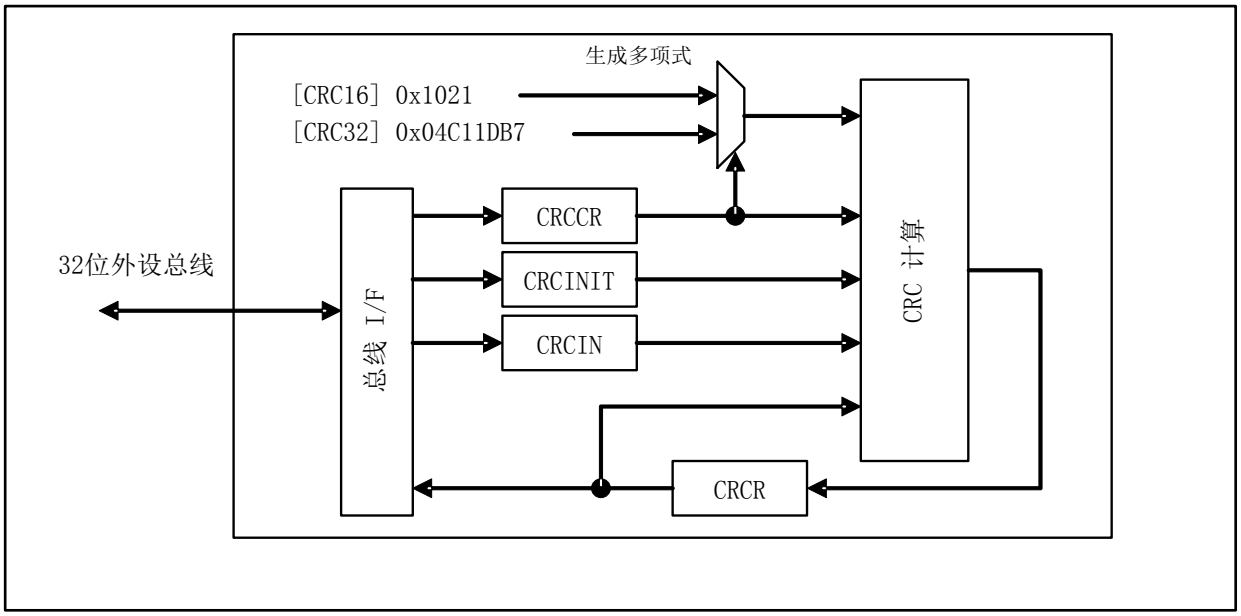
此模块使能 CCITT CRC16 和 IEEE-802.3 CRC32 两种模式进行计算。在此模块中，这两种模式的生成多项式固定为数字值。所以不能计算基于其它生成多项式的 CRC 值。

- CCITT CRC16 生成多项式：0x1021
- IEEE-802.3 CRC32 生成多项式：0x04C11DB7

CRC 框图

Figure 1-1 所示为 CRC 框图。

Figure 1-1CRC 框图



- CRCCR（CRC 控制寄存器）

  用于控制 CRC 计算。

- CRCINIT（CRC 初始值寄存器）

  用于指定 CRC 计算所用的初始值。

- CRCIN（输入数据寄存器）

  用于设置 CRC 计算所用的输入数据。

- CRCR（CRC 寄存器）

  用于输出 CRC 计算结果。

- CRC 计算

  执行 CRC 计算的电路。

2. CRC 操作

本节概述 CRC 操作。

CRC 定义

■ CCITT CRC16 标准

生成多项式	0x1021	(CRCCR:CRC32=0)
初始值	0xFFFF	
XOR 终值	0x0000	(CRCCR:FXOR=0)
位序	高位优先	(CRCCR:LSBFST=0)
输出位序	低位优先	(CRCCR:CRCLSF=0)

（输入-输出字节序可任意指定。）

■ IEEE-802.3 CRC32 以太网标准

生成多项式	0x04C11DB7	(CRCCR:CRC32=1)
初始值	0xFFFFFFFF	
XOR 终值	0xFFFFFFFF	(CRCCR:FXOR=1)
位序	低位优先	(CRCCR:LSBFST=1)
输出位序	低位优先	(CRCCR:CRCLSF=1)

（输入-输出字节序可任意指定。）

复位操作

复位时，初始值寄存器(CRCINIT)和 CRC 寄存器(CRCR)设置为 0xFFFFFFFF。其它寄存器清除为"0"。

初始化

通过初始化位(CRCCR:INIT)进行初始化可将初始值寄存器的值载入 CRC 寄存器(CRCR)。

字节序和位序处理

下面以实例来说明字节序和位序的处理。

将下面这个字输入 CRC 计算单元。

133.82.171.1 = 10000101 01010010 10101011 00000001

当字节序设置为大端(CRCCR:LTLEND=0)时，字节发送序列设定如下：

10000101 01010010 10101011 00000001
(第一) (第二) (第三) (第四)

当位序设置为小端(CRCCR:LSBFST=1)时，位发送序列设定如下：

10100001 01001010 11010101 10000000
(起始位) (结束位)

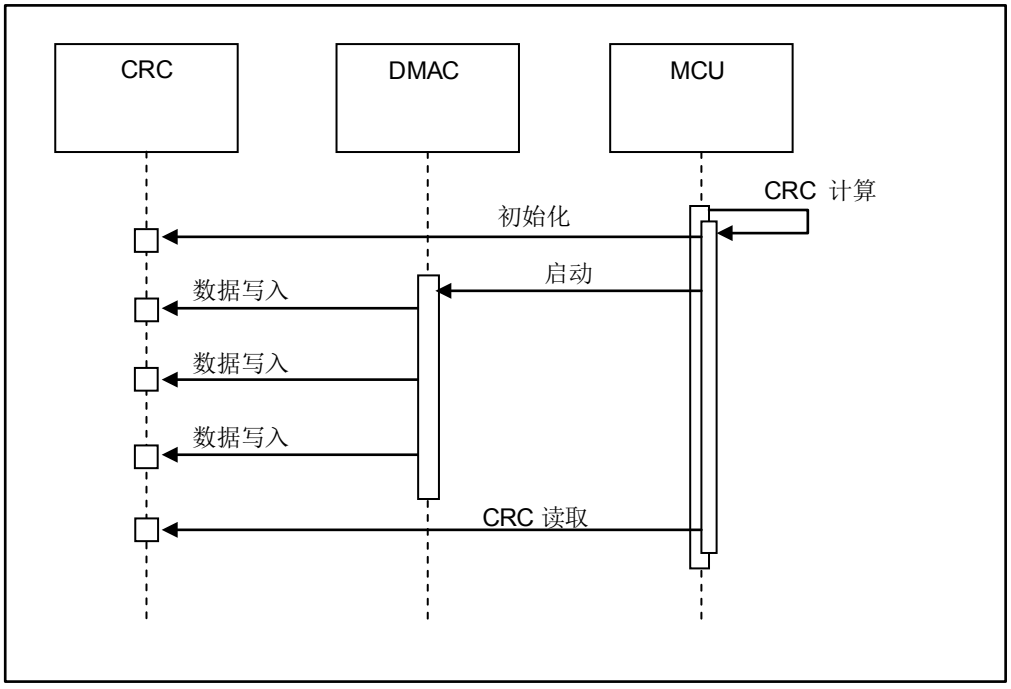
注意事项:

当 CRCCR:CRCLTE=1 时，CRC16 和 CRC32 两种模式的 CRC 结果均采用字节按 32 位宽度重新排列。特别应注意的是，CRC16 模式的数据是输出至 bit 31 ~ bit 16。

2.1 CRC 计算序列

Figure 2-1 所示为 CRC 计算序列。在本节中，假定已配置初始值寄存器(CRCINIT)设置、CRC16/CRC32 模式选择(CRCCR:CRC32)以及字节序或位序设置(CRCCR:LTLEND、CRCCR:LSBFST)。
如果初始值可设置为 0xFFFFFFFF，则可省略初始值寄存器(CRCINIT)设置。

Figure 2-1CRC 计算序列



- 初始化时，初始值位(CRCCR.INIT)写入 1。初始值寄存器(CRCINT)值载入 CRC 寄存器(CRCR)。

- 写入输入数据时，写入数据寄存器(CRCIN)。然后启动 CRC 计算。如有必要，可连续写入输入数据。此外，写入输入数据序列中可采用不同位宽。

- 获取 CRC 代码时，读取 CRC 寄存器 (CRCR)。

2.2 CRC 使用示例

Figure 2-2 至 Figure 2-5 所示为 CRC 使用示例。.

使用示例 1：CRC16，固定字节输入

Figure 2-2 使用示例 1（CRC16，固定字节输入，内核字节序：大端）

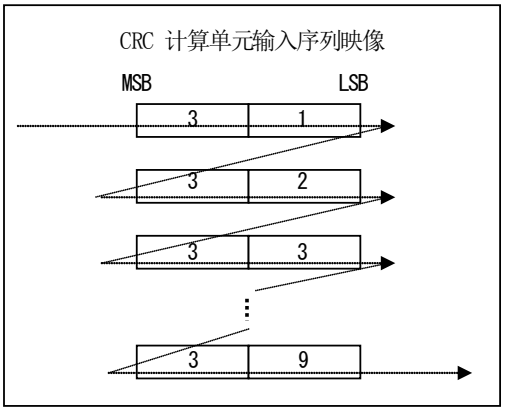
```
//**************************************************
//  CRC16 (CRC ITU-T)
//      多项式:        0x1021
//      初始值:    0xFFFF
//        CRCCR.CRC32:       0 //CRC16
//        CRCCR.LTLEND:      0 //大端
//        CRCCR.LSBFST:      0 //MSB 优先
//        CRCCR.CRCLTE:      0 //CRC 大端
//        CRCCR.CRCLSF:      0 //CRC MSB 优先
//        CRCCR.FXOR:        0 //CRC 最终 XOR 关
//**************************************************

//
// 示例 1-1 基于字节的写入
//

// 初始化
B_WRITE (CRCCR, 0x01);

// data write 0x313233343536373839
B_WRITE (CRCIN, 0x31);
B_WRITE (CRCIN, 0x32);
B_WRITE (CRCIN, 0x33);
B_WRITE (CRCIN, 0x34);
B_WRITE (CRCIN, 0x35);
B_WRITE (CRCIN, 0x36);
B_WRITE (CRCIN, 0x37);
B_WRITE (CRCIN, 0x38);
B_WRITE (CRCIN, 0x39);

// 读取结果
H_READ (CRCR+2, data);

//检验结果
assert (data    == 0x29B1);

//
// 示例 1-2 CRC 检验
//

// 初始化
B_WRITE (CRCCR, 0x01);

// 数据写入 0x313233343536373839 + CRC
B_WRITE (CRCIN, 0x31);
B_WRITE (CRCIN, 0x32);
B_WRITE (CRCIN, 0x33);
B_WRITE (CRCIN, 0x34);
B_WRITE (CRCIN, 0x35);
B_WRITE (CRCIN, 0x36);
B_WRITE (CRCIN, 0x37);
B_WRITE (CRCIN, 0x38);
B_WRITE (CRCIN, 0x39);
B_WRITE (CRCIN, 0x29); // <- CRC
B_WRITE (CRCIN, 0xB1); // <- CRC

//读取结果
H_READ (CRCR+2, data);

// 检验结果
assert (data == 0x0000);
```

(假定如下)

B_WRITE -- 字节写入
H_WRITE -- 半字写入
W_WRITE -- 字写入

B_READ -- 字节读取
H_READ -- 半字读取
W_READ -- 字读取

CRCCR -- CRC 控制寄存器地址
CRCINIT -- 初始值寄存器地址
CRCIN -- 输入数据寄存器地址
CRCR -- CRC 寄存器地址



- 字节和半字写入位置是任意的。在本例中，数据连续写入位置+0。
- Table 2-1 列出了 CRC16 模式下 CPU、CRC 结果的字节序、CRCR (CRC 寄存器)输出位置以及读取地址。

Table 2-1CPU、CRC 结果的字节序及 CRCR 读取地址

内核字节序	CRC 结果字节序	对 CRCR 的输出位置	CRCR H_READ 地址
大端	大端	bit 15~bit 0	CRCR +2
大端	小端	bit 31 ~ bit 16	CRCR +0
小端	大端	bit 15 ~ bit 0	CRCR +0
小端	小端	bit 31 ~ bit 16	CRCR +2

使用示例 2：CRC16，混合不同输入位宽度

Figure 2-3 使用示例 2（CRC16，混合不同输入位宽度，内核字节序：大端）

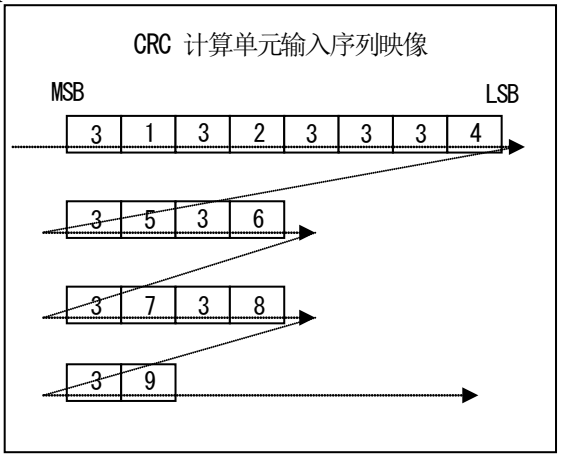
```
//*********************************************
//    CRC16 (CRC ITU-T)
//          多项式:          0x1021
//          初始值:     0xFFFF
//          CRCCR.CRC32:        0 //CRC16
//          CRCCR.LTLEND:       0 //大端
//          CRCCR.LSBFST:       0 //MSB 优先
//          CRCCR.CRCLTE:       0 //CRC 大端
//          CRCCR.CRCLSF:       0 //CRC MSB 优先
//          CRCCR.FXOR:         0 //CRC 最终 XOR 关
//*********************************************

//
// 示例 2-1 混合写入宽度
//

// 初始化
B_WRITE (CRCCR, 0x01);

// data write 0x313233343536373839
W_WRITE (CRCIN, 0x31323334);
H_WRITE (CRCIN, 0x3536);
H_WRITE (CRCIN+2, 0x3738);
B_WRITE (CRCIN+3, 0x39);

//读取结果
H_READ (CRCR+2, data);

// 检验结果
assert (data == 0x29B1);

//
//示例 2-2 CRC 检验
//

//初始化
B_WRITE (CRCCR, 0x01);

// 数据写入 0x313233343536373839 + CRC
W_WRITE (CRCIN, 0x31313334);
W_WRITE (CRCIN, 0x35363738);
H_WRITE (CRCIN, 0x3929);        // <-- CRC(0x29)
B_WRITE (CRCIN, 0xB1);          // <-- CRC(0xB1)

//读取结果
H_READ (CRCR+2, data);

// 检验结果
assert (data == 0x0000);
```

(假定如下)

B_WRITE – 字节写入
H_WRITE – 半字写入
W_WRITE – 字写入

B_READ – 字节读取
H_READ – 半字读取
W_READ – 字读取

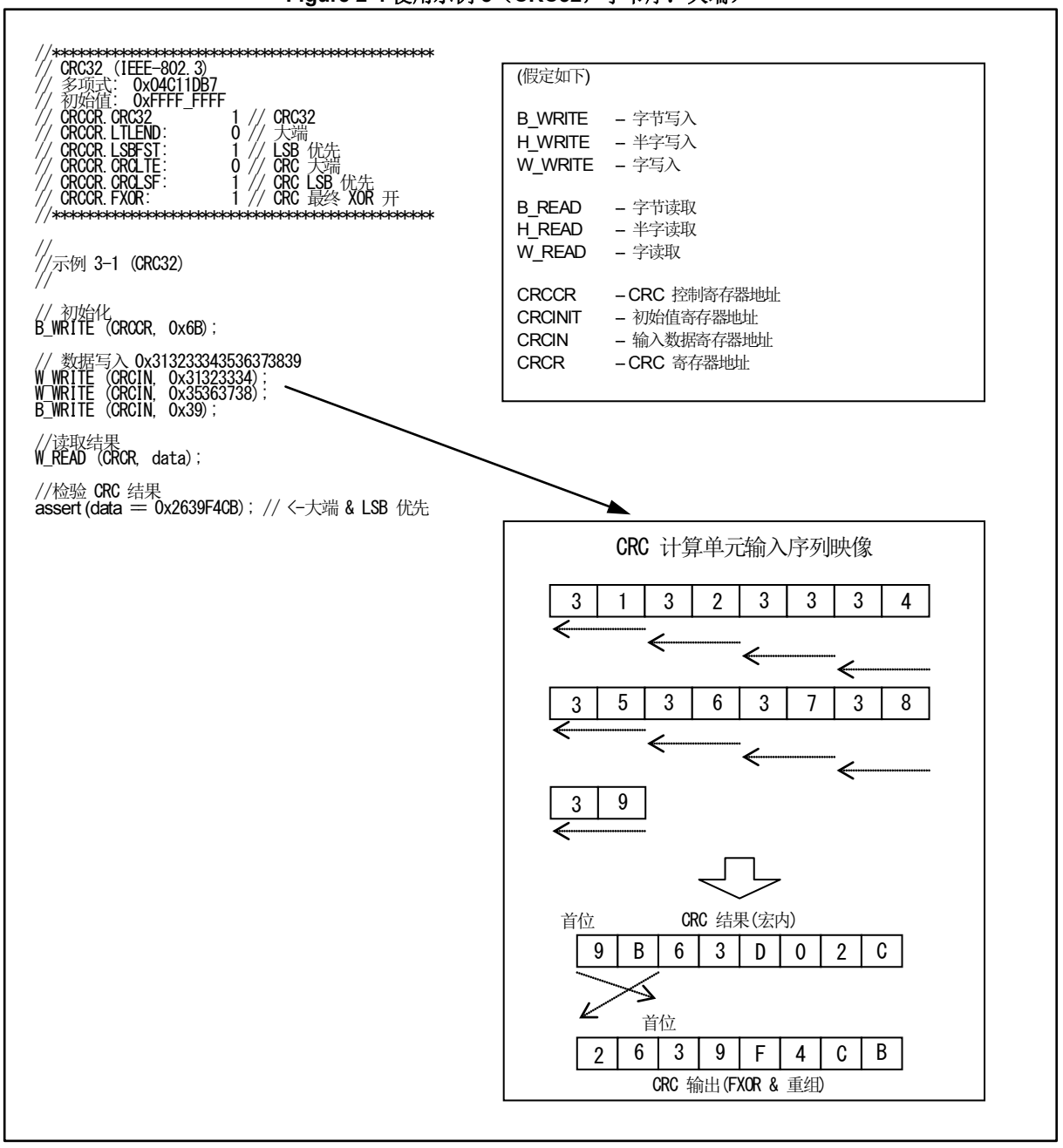
CRCCR – CRC 控制寄存器地址
CRCINIT – 初始值寄存器地址
CRCIN – 输入数据寄存器地址
CRCR – CRC 寄存器地址



– 如果字节序或位序设置正确且至 CRC 计算单元的位输入序列相同，则写入宽度可任意指定。
例如，如果在基于字的写入模式时最终获得的是 1 字节、2 字节或 3 字节分数，则字节和半字写入均使能。

使用示例 3：CRC32，字节序：大端

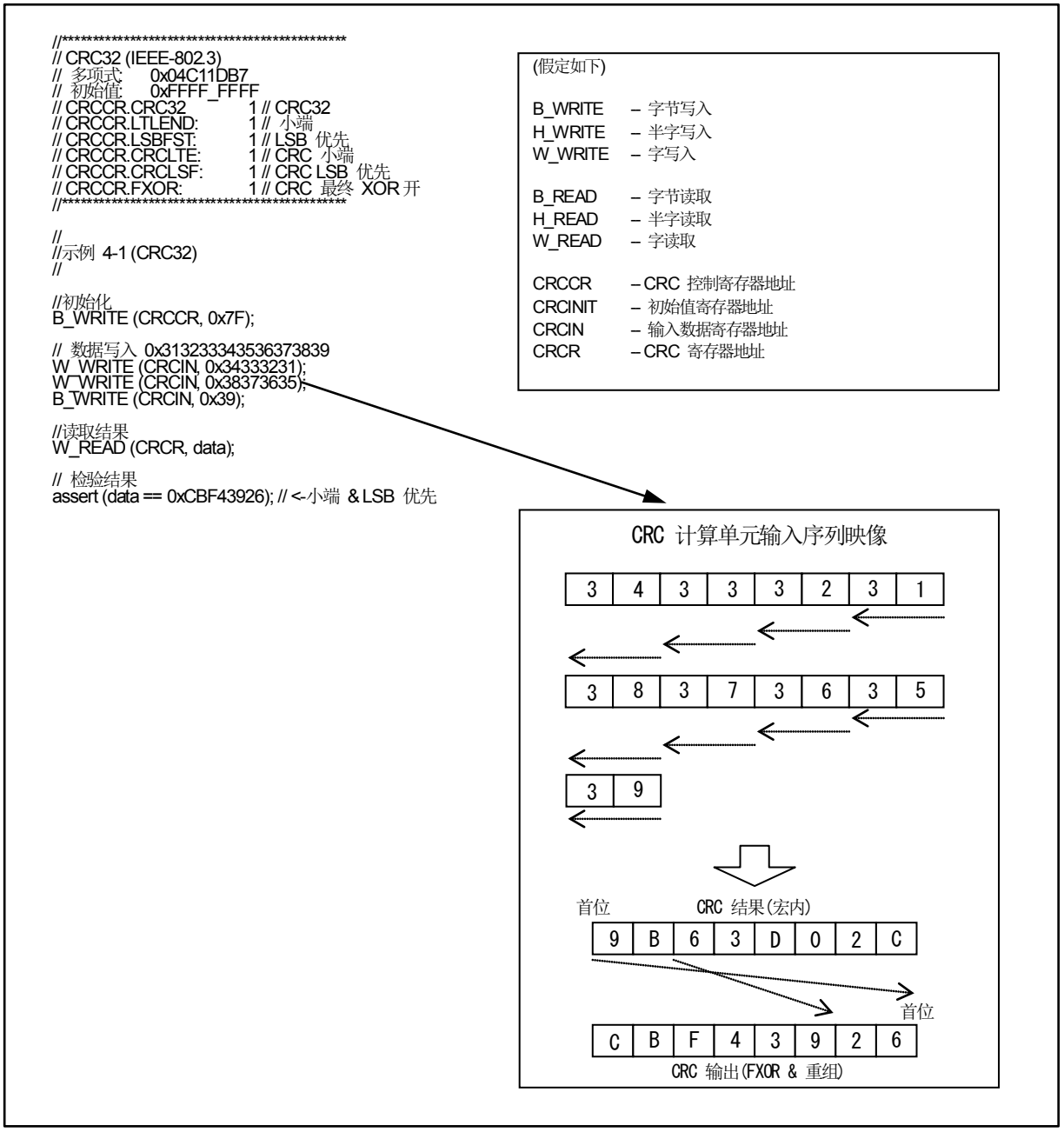
Figure 2-4 使用示例 3（CRC32，字节序：大端）



- 在 CRC32 (IEEE-802.3)模式下，位序设置为 LSB 优先。此 CRC 计算单元支持大端和小端字节序。Figure 2-4 所示为大端示例。

使用示例 4：CRC32，字节序：小端

Figure 2-5 使用示例 4（CRC32，字节序：小端）



- 在 CRC32 (IEEE-802.3)模式下，位序设置为 LSB 优先。此 CRC 计算单元支持大端和小端字节序。Figure 2-5 所示为小端示例。

- 如果 CRC 结果不需要位反转，执行下列进程之一解除对当前结果的位反转。

- 计算之前，通过设置 CRCCR=0x3F(CRCCR:FXOR=0, CRCCR:INIT=1)进行初始化。
- 数据输入之后，设置 CRCCR=0x3E (CRCCR:FXOR=0, CRCCR:INIT=0)。

3. CRC 寄存器

本节提供 CRC 寄存器列表。

CRC 寄存器

Table 3-1CRC 寄存器列表

缩写	寄存器名称	参考章节
CRCCR	CRC 控制寄存器	3.1
CRCINIT	初始值寄存器	3.2
CRCIN	输入数据寄存器	3.3
CRCR	CRC 寄存器	3.4

3.1 CRC 控制寄存器 (CRCCR)

CRC 控制寄存器(CRCCR)用于控制 CRC 计算。

位	7	6	5	4	3	2	1	0
字段	保留	FXOR	CRCLSF	CRCLTE	LSBFST	LTLEND	CRC32	INIT
属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

[bit7]保留：保留位

读取值为 "0"。

确定在本位写入"0"。

[bit6] FXOR：最终 XOR 控制位

此位用于将 CRC 结果输出为 XOR 值或 XOR。

XOR 值设置为 0xFFFFFFFF。当 FXOR="1"时，反转 CRC 结果值。

此项处理是在 CRC 寄存器(CRCR)进程的后半部分进行的。因此，本位设置后 CRC 结果立即会在读取值反映出来。

位	描述
0	无
1	有

[bit5] CRCLSF：CRC 结果位序设置位

本位为 CRC 结果的位序设置位。

本位用于重新排列各字节的位。设置为"0"指定 MSB 优先，设置为"1"指定 LSB 优先。

此项处理是在 CRC 寄存器(CRCR)进程的后半部分进行的。因此，本位设置后 CRC 结果立即会在读取值反映出来。

位	描述
0	高位优先
1	低位优先

[bit4] CRCLTE：CRC 结果字节序设置位

本位为 CRC 结果的字节序设置位。

本位用于重排各字的字节序。设置为"0"指定大端字节序，设置为"1"指定小端字节序。

此项处理是在 CRC 寄存器(CRCR)进程的后半部分进行的。因此，本位设置后 CRC 结果立即会在读取值反映出来。

如果在 CRC16 模式时将本位设置为"1"，数据输出至 CRC 寄存器(CRCR)的 D[31:16]。

位	描述
0	大端
1	小端

[bit3] LSBFST：位序设置位

本位为位序设置位。

本位用于指定字节（8 位）的首位。设置为"0"指定 MSB 优先，设置为"1"指定 LSB 优先。

当本位与 LTLEND 位设置相结合时，可指定四种类型的处理次序。

位	描述
0	高位优先
1	低位优先

[bit2] LTLEND：字节序设置位

本位为字节序设置位。

本位用于指定与写入宽度的字节序。设置为"0"指定大端字节序，设置为"1"指定小端字节序。

位	描述
0	大端
1	小端

[bit1] CRC32：CRC 模式选择位

本位用于选择 CRC16 或 CRC32 模式。

位	描述
0	CRC16
1	CRC32

[bit0] INIT：初始化位

本位为初始化位。本位写入"1"时，数据初始化。本位没有值，读取时总会返回"0"。

初始化时，初始值寄存器(CRCINIT)的值载入 CRC 寄存器(CRCR)。

每次启动 CRC 计算时必须执行初始化。

位	描述	
	写入	读取
0	无操作	总读为"0"。
1	初始化	

3.2 初始值寄存器 (CRCINIT)

初始值寄存器(CRCINIT)用于保存 CRC 计算的初始值。

位	31															0
字段	D[31:0]															
属性	R/W															
初始值	0xFFFFFFFF															

[bit31:0] D [31:0]：初始值位

这些位用于保存 CRC 计算的初始值。

CRC 计算的初始值写入此寄存器。

（复位时 0xFFFFFFFF）

在 CRC16 模式下，使用 D15 ~ D0，忽略 D31 ~ D16。

3.3 输入数据寄存器 (CRCIN)

输入数据寄存器(CRCIN)用于设置 CRC 计算所用的输入数据。

位	31	0
字段	D[31:0]	
属性	R/W	
初始值	0x00000000	

[bit31:0] D[31:0]：输入数据位

这些位用于设置 CRC 计算所用的输入数据。

CRC 计算的输入数据写入此寄存器。有三种位宽：8 位、16 位和 32 位（字节、半字和字）可同时指定。字节和半字写入位置是任意的。可用地址位置说明如下：

字节写入：+0, +1, +2, +3

半字写入：+0, +2

3.4 CRC 寄存器 (CRCR)

CRC 寄存器 (CRCR)用于输出 CRC 计算结果。启动计算前必须初始化寄存器。

位	31															0
字段	D[31:0]															
属性	R															
初始值	0xFFFFFFFF															

[bit31:0] D [31:0]： CRC 位

这些位用于读取 CRC 计算结果。在初始化位(CRCCR:INIT)写入"1"时，初始值寄存器(CRCINIT)的值载入此寄存器。

当 CRC 计算的输入数据写入输入数据寄存器(CRCIN)时，在一个机器时钟周期结束后将 CRC 计算结果设置至此寄存器。当所有输入数据写入完成后，此寄存器保持最终 CRC 代码。

在 CRC16 模式下，如果字节序设置为大端(CRCLTE=0)，结果输出至 D15 ~ D0。当字节序设置为小端(CRCLTE=1)时，结果输出至 D31 ~ D16。

第 12 章：调试接口

本章说明调试接口功能及操作。

1. 概述
2. 引脚描述

代码：9AFDEBUG-C01.0

1. 概述

本产品家族包括串行线调试端口 (SW-DP)。

ICE 连接至 SW-DP 可进行系统调试。

本系列产品还包括有记录**程序流程**变化的微跟踪缓冲 (MTB)功能。

本节说明调试接口。

SW-DP 和系统调试详见《Cortex-M0+技术参考手册》。

功能特性

两个引脚被指派给 SW-DP。

这两个引脚的初始功能为串行线调试。

2. 引脚描述

本节说明引脚。

2.1 调试用引脚
2.2 引脚初始功能
2.3 SW-DP 引脚内部上拉

2.1 调试用引脚

串行线分配两个引脚(SWCLK 和 SWDIO)。

Table 2-1 为引脚功能列表。

Table 2-1 调试模式的 SW-DP 引脚功能

引脚	功能
SWCLK	串行线时钟信号
SWDIO	串行线数据输入/输出信号

2.2 引脚初始功能

两个 SW-DP 引脚也用为 GPIO。
SW-DP 引脚的初始功能为调试。

注意事项： 引脚功能设置，详见“I/O 端口”一章。

Table 2-2 列出清除复位后的初始状态以及可通过设置 PFR（端口功能寄存器）改变的功能。

注意事项： 有关 PFR，详见“I/O 端口”一章。

Table 2-2 调试用引脚初始功能和功能改变

	引脚名称	引脚初始功能	PFR 设置的功能改变
SW-DP 引脚	SWCLK	SWCLK	GPIO
	SWDIO	SWDIO	GPIO

2.3 SW-DP 引脚内部上拉

如 ARM 标准规定，本家族产品提供具有上拉功能的调试引脚。
用户可设置 GPIO 中相应寄存器设置，控制上拉。

Table 2-3 SW-DP 引脚内部上拉使能或禁用状态

引脚名称	调试引脚上拉使能*
SWCLK	使能
SWDIO	使能

*: 复位时也能使能上拉。

第 13 章微跟踪缓冲区数据监视点及跟踪

本章说明 **MTB_DWT**（微跟踪缓冲区数据监视点及跟踪）的功能和操作。

1. **概述**
2. **框图**
3. **配置及设置步骤示例**
4. **寄存器**

代码：9AFEXTINT-C01.0_FW12-E0.14

1. 概述

MTB_DWT 用于产生 ARM 核心视线 MTB 的 TSTOP 和 TSTART 信号。 MTB_DWT 功能用于访问数据相位时监视处理器地址和数据总线，可检测配置监视点，触发 MTB 记录的适当响应。

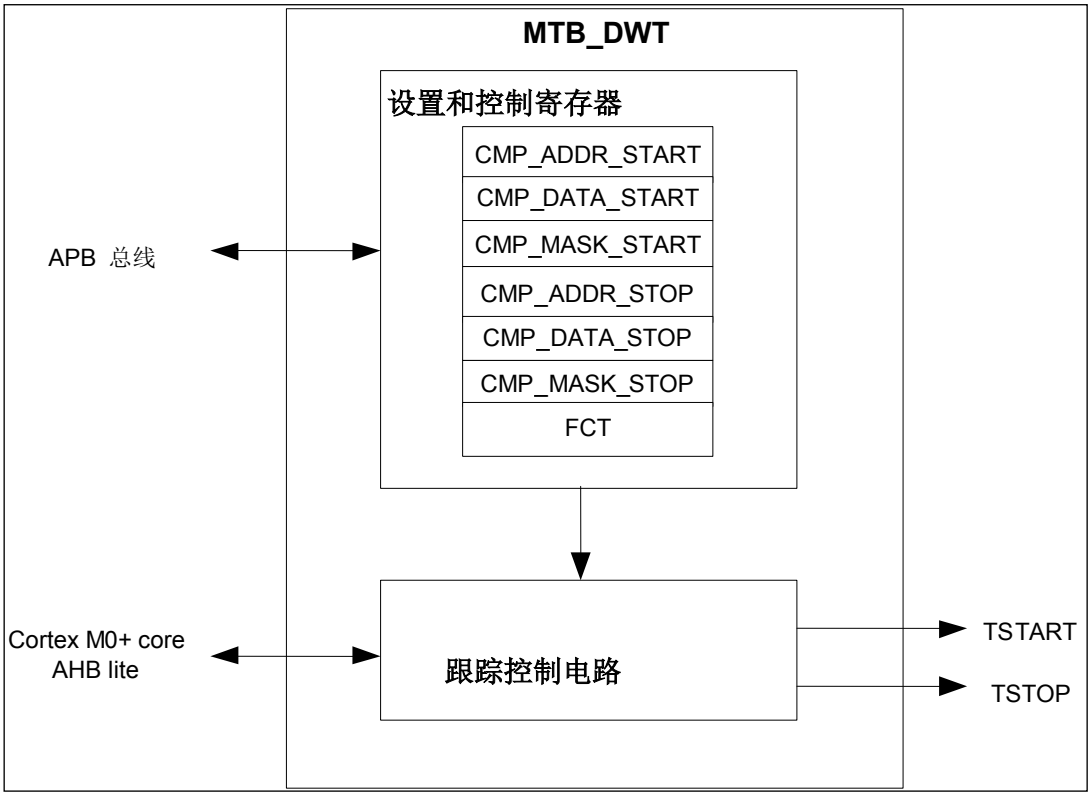
MTB_DWT 功能

- 支持地址以及地址+数据可编程启动/停止记录。
- 支持屏蔽数据各位。
- 支持写入/读取和读取或写入操作监视。
- 支持字节、半字和字监视。
- 支持数据相监视功能。

2. 框图

下面所示为 MTB_DWT 框图。

Figure 2-1MTB_DWT 框图



3. 配置及设置步骤示例

本节说明配置和设置步骤示例。

3.1 MTB_DWT 配置
3.2 设置步骤

3.1 MTB_DWT 配置

本节说明 MTB_DWT 的配置。

MTB_DWT 配置概述

MTB_DWT 只支持小端数据格式。使能字节或半字访问时，MTB_DWT 数据比较启动跟踪寄存器 (CMP_DATA_START) 和 MTB_DWT 数据比较停止跟踪寄存器 (CMP_DATA_STOP)的未使用位如 Table 3-1 所示。表中所标 "❍"表示有效数据，"-" 表示无效数据。
通过设置 CMP_MASK_START/CMP_MASK_STOP 用无效数据屏蔽相应位。

Table 3-1 AHB-Lite 字节通道定义

访问	地址相	对应数据相			
	CMP_ADDRx*1 [1:0]	CMP_DATAx*2 [31:24]	CMP_DATAx*2 [23:16]	CMP_DATAx*2 [15:8]	CMP_DATAx*2 [7:0]
字节	00	-	-	-	❍
	01	-	-	❍	-
	10	-	❍	-	-
	11	❍	-	-	-
半字	00	-	-	❍	❍
	10	❍	❍	-	-
字	00	❍	❍	❍	❍

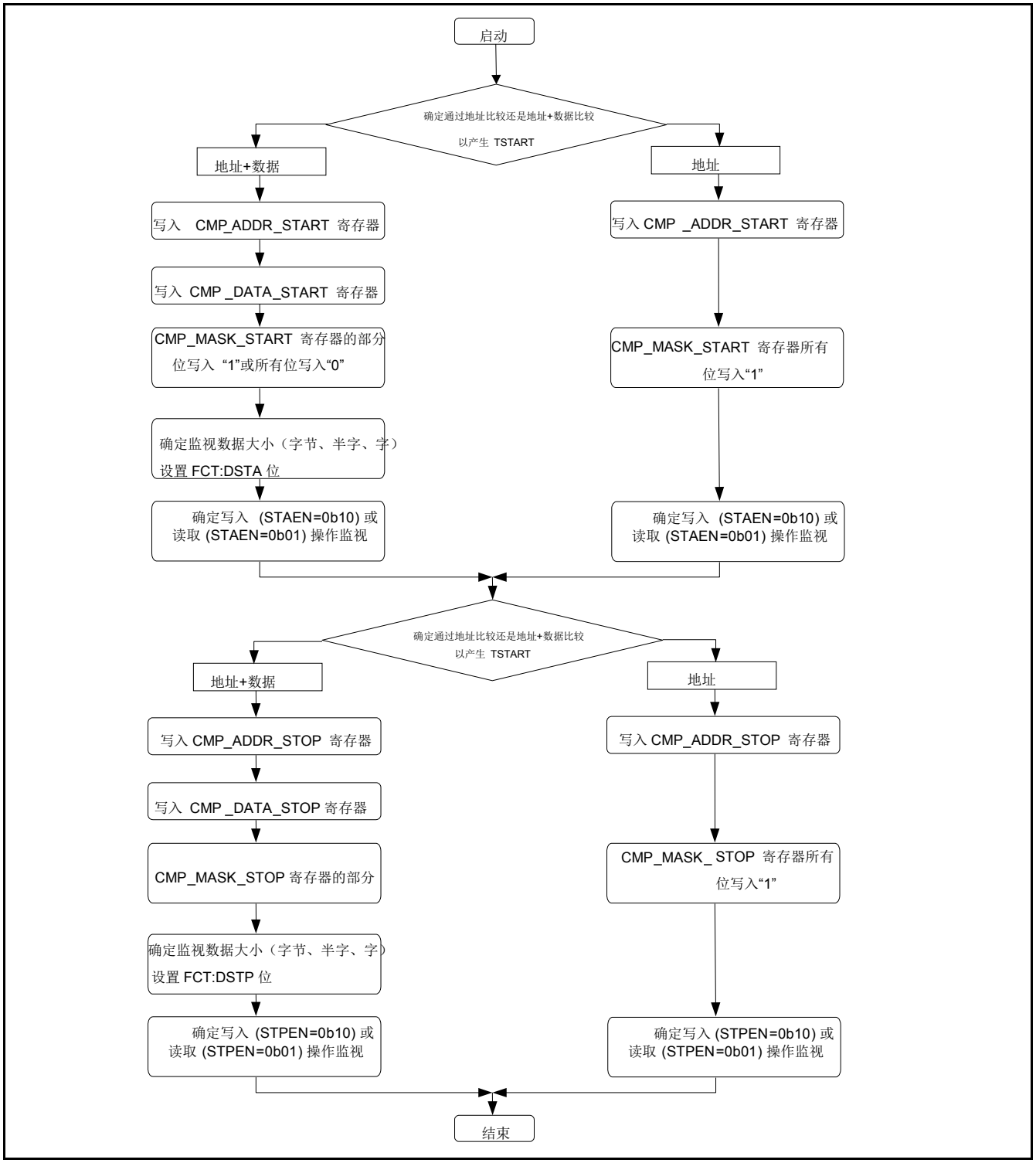
*1: CMP_ADDRx: CMP_ADDR_START 寄存器或 CMP_ADDR_STOP 寄存器。

*2: CMP_DATAx: CMP_DATA_START 寄存器或 CMP_DATA_STOP 寄存器。

3.2 设置步骤

MTB_DWT 设置步骤示例，参见 Figure 3-1。

Figure 3-1 MTB_DWT 设置步骤



4. 寄存器

本节说明 MTB_DWT 寄存器。

MTB_DWT 寄存器列表

Table 4-1 MTB_DWT 寄存器列表

缩写	寄存器名称	参考章节
CMP_ADDR_START	MTB_DWT 地址比较启动跟踪寄存器	4.1
CMP_DATA_START	MTB_DWT 数据比较启动跟踪寄存器	4.2
CMP_MASK_START	MTB_DWT 屏蔽数据比较启动跟踪寄存器	4.3
CMP_ADDR_STOP	MTB_DWT 地址比较停止跟踪寄存器	4.4
CMP_DATA_STOP	MTB_DWT 数据比较停止跟踪寄存器	4.5
CMP_MASK_STOP	MTB_DWT 屏蔽数据比较停止跟踪寄存器	4.6
FCT	MTB_DWT 功能寄存器	4.7
PID4	外设 ID4 寄存器	4.8
PID5	外设 ID5 寄存器	
PID6	外设 ID6 寄存器	
PID7	外设 ID7 寄存器	
PID0	外设 ID0 寄存器	
PID1	外设 ID1 寄存器	
PID2	外设 ID2 寄存器	
PID3	外设 ID3 寄存器	
CID0	元件 ID0 寄存器	4.9
CID1	元件 ID1 寄存器	
CID2	元件 ID2 寄存器	
CID3	元件 ID3 寄存器	

4.1 MTB_DWT 地址比较启动跟踪寄存器 (CMP_ADDR_START)

MTB_DWT 地址比较启动跟踪寄存器 (CMP_ADDR_START)提供产生启动触发信号 TSTART 的基准地址值。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	ADCMP_STA[31:16]															
属性	R/W															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	ADCMP_STA[15:0]															
属性	R/W															
初始值	0x0000															

寄存器功能

[bit31:0] ADCMP_STA [31:0]： MTB_DWT 地址比较启动跟踪位

地址比较基准值用于生成 MTB 启动触发。

注意事项：

- *进行字访问时，bit[3:0]必须设置为下列之一：0x0、0x4、0x8 和 0xC。*
- *进行半字访问时，bit[1:0]必须设置为 0x0 和 0x2。*
- *此寄存器可通过 PRESET0 清除。经 PRESET0 后，此寄存器应重新配置。*

4.2 MTB_DWT 数据比较启动跟踪寄存器 (CMP_DATA_START)

MTB_DWT 数据比较启动跟踪寄存器 (CMP_DATA_START) 提供生成启动触发信号 TSTART 的基准数据值。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	DTCMP_STA[31:16]															
属性	R/W															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	DTCMP_STA[15:0]															
属性	R/W															
初始值	0x0000															

寄存器功能

[bit31:0] DTCMP_STA [31:0]： MTB_DWT 数据比较启动跟踪位

数据比较基准值用于生成 MTB 启动触发。

当所有位都被 CMP_MASK_START 屏蔽时， MTB TSTART 信号仅由比较地址结果确定；否则，MTB TSTART 信号由数据 CMP_DATA_START 和 CMP_DATA_START 比较结果确定。

注意事项：

- *进行字节或半字访问时，按照 Table 3-1 对 CMP_MASK_START 寄存器进行设置，忽略 CMP_DATA_START 寄存器中的未使用位。*
- *此寄存器可通过 PRESET0 清除。经 PRESET0 后，此寄存器应重新配置。*

4.3 MTB_DWT 屏蔽数据比较启动跟踪寄存器 (CMP_MASK_START)

MTB_DWT 屏蔽数据比较启动跟踪寄存器 (CMP_MASK_START) 用于定义 MTB_DWT 数据比较启动跟踪寄存器(CMP_DATA_START) 的忽略位。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	MSK_STA[31:16]															
属性	R/W															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	MSK_STA [15:0]															
属性	R/W															
初始值	0x0000															

寄存器功能

[bit31:0] MSK_STA [31:0]：MTB_DWT 数据比较启动跟踪寄存器屏蔽位

此寄存器屏蔽启动 MTB 的基准数据值。

位	功能
0	操作无效 [初始值]
1	屏蔽对应位

注意事项：

- *当此寄存器的所有位都设置为“1”时，CMP_DATA_START 寄存器的值以及 FCT 寄存器 DSTA 位的值被忽略。*
- *此寄存器可通过 PRESET0 清除。经 PRESET0 后，此寄存器应重新配置。*

4.4 MTB_DWT 地址比较停止跟踪寄存器 (CMP_ADDR_STOP)

MTB_DWT 地址比较停止跟踪寄存器 (CMP_ADDR_STOP) 提供生成停止触发信号 TSTOP 的基准地址值。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	ADCMP_STO[31:16]															
属性	R/W															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	ADCMP_STO[15:0]															
属性	R/W															
初始值	0x0000															

寄存器功能

[bit31:0] ADCMP_STO [31:0]： MTB_DWT 地址比较停止跟踪位

地址比较基准值用于生成 MTB 停止触发。

注意事项：

- *进行字访问时，bit[3:0]必须设置为下列之一：0x0、0x4、0x8 和 0xC。*
- *进行半字访问时，bit[1:0]必须设置为 0x0 和 0x2。*
- *此寄存器可通过 PRESET0 清除。经 PRESET0 后，此寄存器应重新配置。*

4.5 MTB_DWT 数据比较停止跟踪寄存器 (CMP_DATA_START)

MTB_DWT 数据比较停止跟踪寄存器 (CMP_DATA_STOP) 提供生成启动触发信号 TSTOP 的基准数据值。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	DTCMP_STO[31:16]															
属性	R/W															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	DTCMP_STO[15:0]															
属性	R/W															
初始值	0x0000															

寄存器功能

[bit31:0] DTCMP_STO [31:0]：MTB_DWT 数据比较停止跟踪位

数据比较基准值用于生成 MTB 停止触发。

当所有位都被 MTB_DWT 屏蔽数据比较停止跟踪寄存器屏蔽时，MTB TSTOP 信号只由比较地址结果确定；否则 MTB TSTOP 信号由数据（MTB_DWT 数据比较停止跟踪寄存器）和地址（MTB_DWT 地址比较停止跟踪寄存器）比较结果确定。

注意事项：

- *进行字节或半字访问时，按照 Table 3-1 对 CMP_MASK_STOP 寄存器进行设置，不忽略 CMP_DATA_STOP 寄存器中的未使用位。*
- *此寄存器可通过 PRESET0 清除。经 PRESET0 后，此寄存器应重新配置。*

4.6 MTB_DWT 屏蔽数据比较停止跟踪寄存器 (CMP_MASK_STOP)

MTB_DWT 屏蔽数据比较停止跟踪寄存器 (CMP_MASK_STOP) 用于定义 MTB_DWT 数据比较停止跟踪寄存器 (CMP_DATA_STOP)的忽略位。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	MSK_STO[31:16]															
属性	R/W															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	MSK_STO[15:0]															
属性	R/W															
初始值	0x0000															

寄存器功能

[bit31:0] MSK_STO [31:0]： MTB_DWT 数据比较停止跟踪寄存器屏蔽位

此寄存器用于屏蔽 MTB 停止的基准数据值。

位	功能
0	操作无效[初始值]
1	屏蔽对应位。

注意事项：

- *当此寄存器的所有位都设置为“1”时，CMP_DATA_STOP 寄存器的值以及 FCT 寄存器 DSTP 位的值会被忽略。*
- *此寄存器可通过 PRESET0 清除。经 PRESET0 后，此寄存器应重新配置。*

4.7 MTB_DWT 功能寄存器 (FCT)

MTB_DWT 功能寄存器 (FCT) 控制读取/写入操作和数据量。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	保留															
属性	-															
初始值	0x0000															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	保留								DSTP		DSTA		STPEN		STAEN	
属性	-								R/W		R/W		R/W		R/W	
初始值	00000000								00		00		00		00	

寄存器功能

[bit31:8] Reserved: Reserved bits

读取值为 “0”。写入模式时无效。

[bit7:6] DSTP：数据量停止位

这些位用于定义停止 MTB 功能的数据值容量。

bit7	bit6	功能
0	0	字节 [初始值]
0	1	半字
1	0	字
1	1	保留

[bit5:4] DSTA：数据量启动位

These bits define the data value size to start MTB function.

bit5	bit4	功能
0	0	字节 [初始值]
0	1	半字
1	0	字
1	1	保留

[bit3:2] STPEN：使能 MTB_DWT 停止 MTB 功能位

这些位用于使能 MTB_DWT 停止 MTB 功能

bit3	bit2	功能
0	0	禁用 MTB_DWT 停止 MTB 功能。[初始值]
0	1	停止 MTB 的数据读取操作。
1	0	停止 MTB 的数据写入操作。
1	1	停止 MTB 的数据读/写操作。

[bit1:0] STAEN：使能 MTB_DWT 启动 MTB 功能位

这些位用于使能 MTB_DWT 启动 MTB 功能

bit1	bit0	功能
0	0	禁用 MTB_DWT 启动 MTB 功能。[初始值]
0	1	启动 MTB 的数据读取操作。
1	0	启动 MTB 的数据写入操作。
1	1	启动 MTB 的数据读/写操作。

注意事项：

- *此寄存器可通过 PRESET0 清除。经 PRESET0 后，此寄存器应重新配置。*

4.8 外设 ID0-7 寄存器 (PID0-7)

外设 ID0-7 寄存器 (PID0-7) 用于指示外设 ID。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	PERID[31:16]															
属性	R															
初始值	0xXXXX															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	PERID[15:0]															
属性	R															
初始值	0xXXXX															

寄存器功能

[bit31:0] PERID [31:0]：外设 ID 位

这些位在外部调试代理自发现进程时硬连线至所用具体值。

位	功能
读取时	读取指定值。 PID0: 0x00000016 PID1: 0x00000048 PID2: 0x00000008 PID3-7: 0x00000000
写入时	操作无效

4.9 元件 ID0-3 寄存器 (CID0-3)

元件 ID0-3 寄存器 (CID0-3) 用于指示元件 ID。

寄存器配置

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
字段	CPNTID[31:16]															
属性	R															
初始值	0xXXXX															

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
字段	CPNTID[15:0]															
属性	R															
初始值	0xXXXX															

寄存器功能

[bit31:0] CPNTID [31:0]：元件 ID 位

这些位在外部调试代理自发现进程时硬连线至所用具体值。

位	功能
读取时	读取指定值。 CID0: 0x0000000D; CID1: 0x00000090; CID2: 0x00000005; CID3: 0x000000B1.
写入时	操作无效

第 14 章：闪存

有关闪存，参见所用产品的《闪存编程手册》。

代码： 9xFLASHTOP-C01.1

第 15 章：唯一 ID 寄存器

本章说明唯一 ID 寄存器功能及操作。

1. 概述
2. 寄存器

代码：9BFUNIQID-C01.0

1. 概述

此功能概述如下：

41 位预设设备唯一值已设置至唯一 ID 寄存器。

将这些位用于加密及产品串行号等不同用途的设备中，这些值各不相同。

此寄存器为只读时，用户不能写入。 同样，这些值也不会因复位或上电/掉电而改变。

2. 寄存器

寄存器配置及功能说明如下：

寄存器列表

缩写名称	寄存器名称	参考章节
UIDR0	唯一 ID 寄存器 0	2.1
UIDR1	唯一 ID 寄存器 1	2.2

2.1 唯一 ID 寄存器 0(UIDR0: 唯一 ID 寄存器 0)

唯一 ID 寄存器 0 说明如下：

位	31	30	29	28	27	26	25	24
字段	UID[27:20]							
属性	R							

位	23	22	21	20	19	18	17	16
字段	UID[19:12]							
属性	R							

位	15	14	13	12	11	10	9	8
字段	UID[11:4]							
属性	R							

位	7	6	5	4	3	2	1	0
字段	UID[3:0]				保留			
属性	R				-			

[bit31:4] UID [27:0]: 唯一 ID 27 至唯一 ID 0
唯一 ID 的 bit27~ bit0。

[bit3:0] 保留：保留位
保留位。读取值无效。

2.2 唯一 ID 寄存器 1(UIDR1: 唯一 ID 寄存器 1)

唯一 ID 寄存器 1 说明如下：

位	31	30	29	28	27	26	25	24
字段	保留							
属性	-							

位	23	22	21	20	19	18	17	16
字段	保留							
属性	-							

位	15	14	13	12	11	10	9	8
字段	保留			UID[40:36]				
属性	-			R				

位	7	6	5	4	3	2	1	0
字段	UID[35:28]							
属性	R							

[bit31:13] 保留：保留位

保留位。读取值无效。

[bit12:0] UID [40:28]: 唯一 ID 40 至唯一 ID 28

唯一 ID 的 bit40 ~ bit28。

第 16 章：DSTC

本章详细描述 **DSTC**（描述符系统数据传输控制器）。

1. **DSTC 概述**
2. **DSTC 操作概述及 DSTC 系统配置**
3. **DSTC 功能及操作**
4. **DSTC 操作和控制示例**
5. **DSTC 寄存器和描述符**

代码：9BFDSTC-FM0-C03.0

1. DSTC 概述

本章概述 DSTC（描述符系统数据传输控制器）。

概述

DSTC（描述符系统数据传输控制器）与 DMAC 类似，是可绕过 CPU 高速传输数据的功能块。使用描述符（以下称 DES）系统方法，DSTC 可以根据使用一个 DES 在内存上创建的指定内容来直接使用内存或者是外设设备，并执行数据传输操作。

一个 DES 指定一组传输控制参数（基本传输设置、传输数量、传输源地址和传输目标地址）。DSTC 可单独倍增 DES，并且最多能建立 1024 条传输通道。

可使用以下三种方法启动数据传输操作： CPU 直接启动（软件启动）；外设设备发出中断信号启动（硬件启动）；以及链路启动功能。

链路启动功能根据当前的 DES 执行传输，然后根据后续的 DES 启动新的传输或者再次根据当前 DES 启动新的传输。可在 DES 内指定是否使用链路启动功能。使用链路启动功能，DSTC 可将多个 DES 指定的其他传输类型在启动 DES 中合并为单个启动触发信号（软件启动/硬件启动），同时执行多种类型的传输。此外，DSTC 也可将 DES 中指定的一项传输操作划分为几项传输操作来执行。

DSTC 具有传输地址和传输数量计数器两种重载功能（在传输期间，内部重载功能可恢复启动传输时的值，在传输结束时，外部重载功能可恢复启动传输时的值）。这两种重载功能有助于控制重复相同的传输操作。

DSTC 可将传输操作的正常结束或异常结束以中断方式通知 CPU。

DSTC 可控制待机模式（低电耗模式）如何停止内部时钟。

DSTC 有独立于 CPU 总线的专用总线，使其具有能在 CPU 总线被访问时还能执行传输操作的配置。

专用总线的配置符合系统总线 (AHB)，可支持 32 位地址空间 (4 GB)。

DSTC 硬件传输支持的通道数量

对于配备 DSTC 的产品，如 DSTC 支持 256 条通道，可使用通道 0 至通道 255 的所有硬件传输通道。如 DSTC 支持 128 条通道，则不能使用通道 128 至通道 255。如 DSTC 支持 64 条通道，则不能使用通道 64 至通道 255。

2. DSTC 操作概述及 DSTC 系统配置

本节概述 DSTC 操作并解释 DSTC 系统配置。

2.1 DSTC 操作概述

DES 系统

DSTC 根据 CPU 在内存上建立的 DES 所指定内容执行传输操作。如 Table 2-1 所示，DES 包含七种设置，即 DES0 至 DES6。这些设置指定基本传输设置，如各区域的传输数量、传输源地址及传输目标地址。（有关各 DES 的位分配参数，参见 "5 DSTC 寄存器和描述符"。）这些设置作为单组形式储存在内存中。DES0 至 DES6 都是 32 位（1 字）。DES4 至 DES6 为可选 DES。可不指定其设置，具体取决于传输的内容。

Table 2-1DES 类型及设置

存储地址	名称	参数
DESP+0x00（固定）	DES0	设置传输的基本设置。
DESP+0x04（固定）	DES1	设置传输的数量。
DESP+0x08（固定）	DES2	设置传输启动时的传输源地址 (SA)。
DESP+0x0C（固定）	DES3	设置传输结束时的传输目标地址 (DA)。
DESP+0x10 - （可变）	DES4	传输结束时控制 DES1 的外部重载。
	DES5	传输结束时控制 DES2 的外部重载。
	DES6	传输结束时控制 DES3 的外部重载。

Figure 2-1 说明了 DSTC 的 DES 系统方法的配置。使用 DSTC 时，要预留足够的自由储存区域，用于储存 DES。因 DSTC 须引用并更新 DES，所以要选择可读可写的存储区域。使用 DSTC 的 DESTP（DES 顶端地址）寄存器设置（Figure 2-1 中的步骤 1）存储区的启动地址。多个 DES 可分配到 DESTP 开始的 4096 字 (16 KB) 区域。该区域最多可分配 1024 个 DES。DSTC 根据从 DESTP 开始的 DES0 区相应地址值（DESP：DES 指针）识别 DES。

传输启动

将传输信息储存在 DES（Figure 2-1 中的步骤 2）内后，可采用以下三种启动触发方法之一启动 DSTC 传输。

- 软件启动（SW 启动）

  软件启动是通过 CPU 直接启动 DSTC 传输的启动触发方法。将所用 DES 的 DESP 写入 SWTR（软件触发信号）寄存器（Figure 2-1 中的步骤 3），启动传输。软件启动被简称为 SW 启动，下文将通过 SW 启动的 DSTC 传输称为 SW 传输。

- 硬件启动（HW 启动）

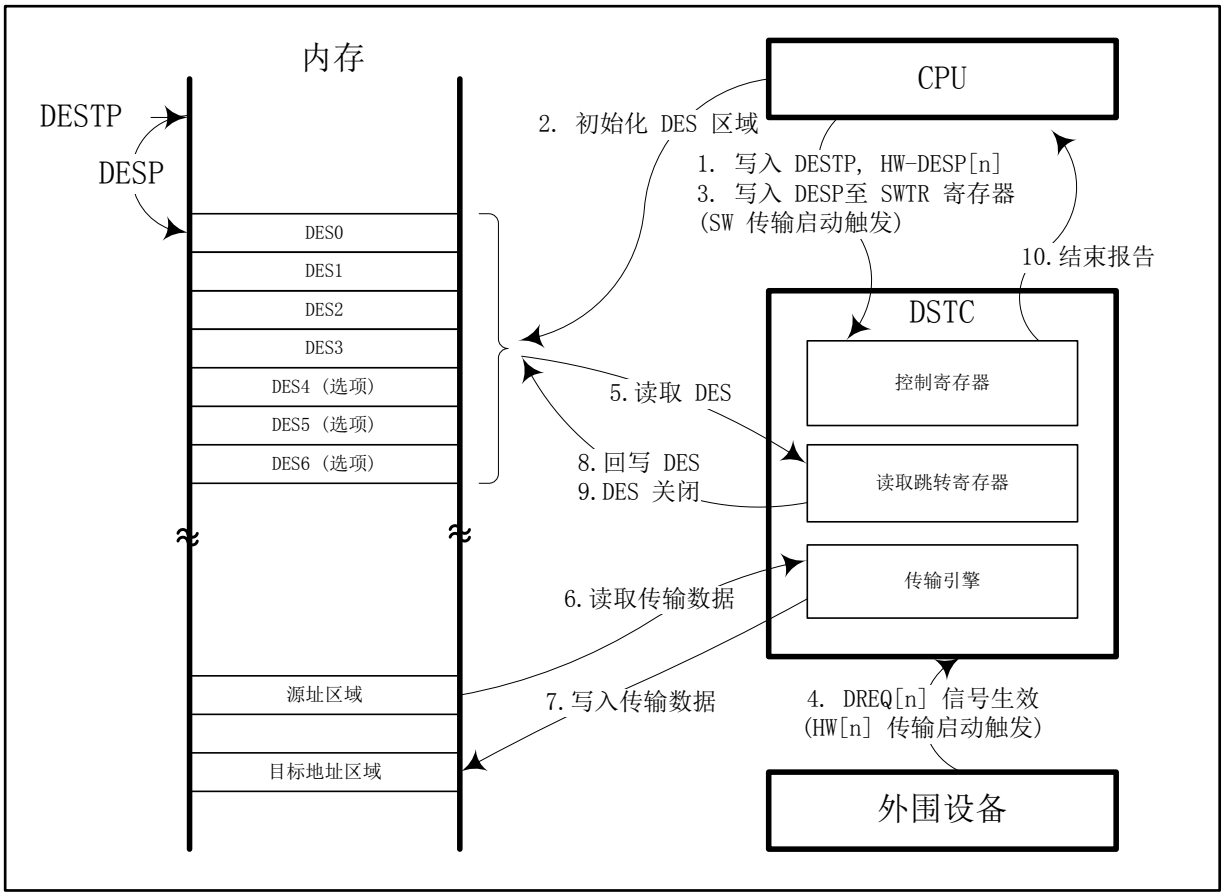
  硬件启动是将外设发出的中断信号用作传输请求信号以启动 DSTC 传输的启动触发方法。外设的中断信号生效时（Figure 2-1 中的步骤 4），系统绕过 CPU 启动 DSTC 传输。预先将此次传输中 DES 的 DESP 写入与 DSTC 内 HW 通道相对应的 HWDESP 寄存器（硬件 DESP）（Figure 2-1 中的步骤 1）。硬件启动简称为 HW 启动，下文将 HW 启动的 DSTC 传输称为 HW 传输。

- 链路启动

  链路启动为 DES 中描述的启动触发方法。 DES 中的传输结束后，如 DES 中有链路启动触发信号，DSTC 将根据后续 DES（或相同的 DES）启动新的传输。可根据当前 DESP 自动计算出链路传输 DES 的新 DESP。

以下章节中，“启动触发信号”指上述所有启动触发信号，包括 SW 启动、HW 启动及链路启动。

Figure 2-1DES 系统方法配置图



传输操作

DSTC 根据上述启动触发信号引用 DESTP+DESP 中的 DES（Figure 2-1 中的步骤 5）。DSTC 检验（DES 打开检验）其引用 DES 的参数，如参数无问题则执行传输（Figure 2-1 中的步骤 6 和 7）。此外，如 DES 具有链路启动触发信号，DSTC 将根据链路启动触发信号执行传输数据。

通过一个启动触发信号执行的传输数量根据 DES 及链路触发信号的参数而变化。不是一个启动触发信号就能结束 DES 指定的所有传输。如发生这种情况，重新将传输数量及更新的传输地址反写至 DES（Figure 2-1 中的步骤 8）。DSTC 等待下一个启动触发信号，并在收到下一个启动触发信号后继续执行传输。

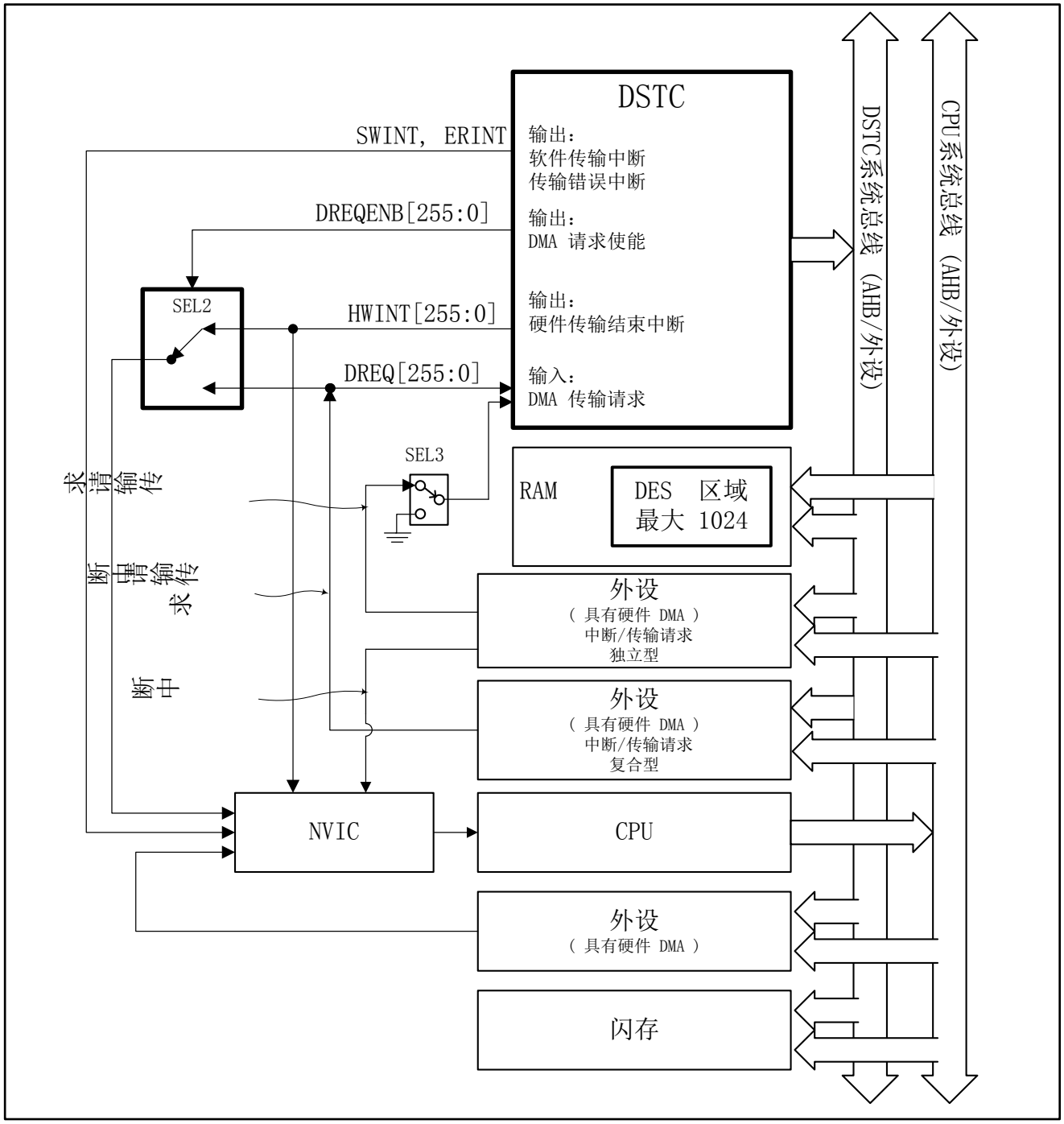
传输结束

DES 中指定的传输全部结束后，DSTC 执行 DES 关闭进程（Figure 2-1 中步骤 9）。DES 设置可阻止 DES 关闭进程的执行。DSTC 可通过中断通知 CPU 以下事项：i. DSTC 等待启动触发信号；ii. 传输正常结束；iii. 传输异常结束。（Figure 2-1 中步骤 10）

2.2 DSTC 和系统配置

Figure 2-2 所示为 DSTC 及系统配置的框图。

Figure 2-2DSTC 及系统配置框图



系统连接

为方便说明，Figure 2-2 为简化系统配置图。有关系统配置的详细说明，参见《外设手册》中“系统概述”一章。 DSTC 通过系统总线连接至 CPU、闪存、RAM 及外设。DSTC 配有独立于 CPU 总线的专用总线，并能在访问 CPU 总线时执行传输操作。DSTC 根据指定的通道传输目标地址和传输源地址访问系统上的所有地址区域，并执行内存与外设之间的数据传输。DSTC 不能访问某些区域。有关 DSTC 不能访问的区域，参见内存映射图。

DREQENB[n] 寄存器及与 DREQ[n] 信号和 HWINT[n] 信号之间的连接

DSTC 最多可支持 256 个硬件传输请求信号输入。支持 DSTC 硬件传输的外设中断信号连接至 DSTC。DSTC 可将外设中断信号用作 DMA 传输请求信号 (DREQ[255:0]) 启动传输操作。DSTC 不能通过不支持 DSTC 硬件传输的外设所发出中断信号启动 DSTC 的 DMA 传输。如果是有多条通道和多项中断源的外设，则会有支持以及不支持 DMA 传输的中断。

DREQENB[255:0] 的设置决定外设的硬件传输请求是否有效。配备 DSTC 的产品规格决定了 256 位中哪个位支持哪种外设的中断信号。有关详细信息，参见“中断”一章。

有两种外设功能使用 DSTC 的 DMA 传输；一种是组合复用中断的输出信号以及 DMA 传输请求信号，另一种则分别处理。DREQENB[255:0] 寄存器的设置值及此图中 SE2 和 SEL3 的转换操作如下：

- 如果外设类型为组合中断和传输请求：

当 DREQENB[n]=0 时；

外设中断被输入至 NVIC，通知中断。
DSTC 忽略外设发出的中断。
DSTC 发出的 HWINT[n] 信号不输入 NVIC。HWINT[n] 为 DSTC 发至 CPU 的中断输出信号，用于通知 DSTC 传输完成。

当 DREQENB[n]=1 时；

外设发出的中断不输入 NVIC。
外设中断输入至 DSTC，且 DSTC 根据此信号启动 DMA 传输。
输入 NVIC 的是 DSTC 发出的 HWINT[n] 信号而非外设发出的中断信号。

这种情况下，外设中断以及 DSTC 发出的 HWINT[n]传输完成中断复用 NVIC 的输入端口。此配置的 NVIC 进程中，外设中断和 DSTC 的传输完成中断跳转至同一中断向量。因此，使用 DREQENB[n] 寄存器选择待处理的中断。

- 如果外设类型为分别处理中断和传输请求：

当 DREQENB[n]=0 时；

外设中断被输入至 NVIC，通知中断。
外设发出的传输请求不输入至 NVIC
DSTC 不忽略外设的传输请求
DSTC 的 HWINT[n]未生效。

当 DREQENB[n]=1 时；

外设中断被输入至 NVIC，通知中断。
外设发出的传输请求不输入至 NVIC
外设的传输请求输入 DSTC，启动传输。
DSTC 发出的 HWINT[n] 输入 NVIC，通知传输完成。

这种情况时，NVIC 输入端口分别处理外设的中断及 DSTC 的传输完成 HWINT[n] 中断。NVIC 进程中，外设的中断及 DSTC 的传输完成中断不会跳转至同一中断向量。

有关外设类型的详细信息，参见“中断”一章中输入 DSTC 的中断列表及中断信号列表。

硬件传输请求清除信号的连接

有的支持硬件传输的外设在传输结束后必须清除传输请求信号（中断信号）。虽然 Figure 2-2 未说明清除进程，如果这种外设的中断请求信号通过对应 DREQENB[n] 寄存器使能，则由 DSTC 清除传输请求信号。

硬件传输停止请求信号的连接

多功能串行单元（本文档以下称 MFS）及 I^2S 输出 DMA 传输停止请求的信号。尽管 Figure 2-2 未说明，如这些外设的传输停止请求信号已生效，则屏蔽传输信号。所以，在等待 DMA 请求信号的状态下 DSTC 不会执行 DMA 传输，且 DSTC 也不会产生错误响应。

MFS 的传输停止请求信号生效条件如下：

−如使能接收到的中断 (SCR:RIE=1)，则发生接收到的中断（SSR:PE 位、FRE 位或 ORE 位设置为 1）。

−如使能芯片选择错误中断 (SACSR:CSEIE=1)，则发生芯片选择错误中断（SACSR:CSE 位设置为 1）。

I2S 的传输停止请求信号生效条件如下所述：

−如通过设置 OPRREG.START 为 0 停止 I^2S，在 DMA 传输请求至传输完成期间。

−如通过设置 OPRREG.RXENB 为 0 禁用接收操作，在 DMA 传输请求至传输完成期间。

−如通过设置 OPRREG.TXENB 为 0 禁用转换操作，在 DMA 传输请求至传输完成期间。

−如通过设置 SRST.SRST 为 1 发生 I2S 软件复位，在 DMA 传输请求至传输完成期间。

−如果 DSTC 唤醒 ACK，在 DMA 传输请求至传输启动期间。

另外，传输停止请求信号作为中断通过 NVIC 发送至 CPU。CPU 通过此中断终止 DSTC 的当前 DMA 传输。有关详细信息，参见“中断”一章。

DSTC 的中断信号

SWINT 将软件启动传输的传输结束中断发送至 NVIC。ERINT 将发生传输错误时产生的错误中断发送至 NVIC。

3. DSTC 功能及操作

本节说明 DSTC 操作。

3.1. DES 设置
3.2. DSTC 的控制功能
3.3. DSTC 操作流程

3.1 DES 设置

本节说明 DES 的详细设置及 DSTC 操作。

3.1.1 传输数据量的指定

TW、IRM、IIN、ORM

DSTC 以单次传输方式传输 DES0 中 TW 指定数据位宽的数据。DSTC 内有传输量计数器。计数器为双循环配置，由外循环计数器和内循环计数器组成。DES1 内 ORM（外循环余值）指外循环计数器的剩余传输数量，DES1 内的 IRM（内循环余值）指内循环计数器的剩余传输数量。DES1 内的 IIN（内循环初值）指定内循环计数器的初始值。

启动传输时，指定传输数据宽以及 TW、IRM、IIN 和 ORM 中的不同传输数量。对于一个 DES，DSTC 传输整个 TW×IIN×ORM 的数据。

Table 3-1 列出指定传输数据宽、传输数量和传输模式的方法。

Table 3-1 指定传输模式、传输大小和传输数量

区域名称	名称	参数
DES0	MODE	MODE 选择传输模式。 0: 选择模式 0。 1: 选择模式 1。
	TW[1:0]	TW 指定单次传输的数据宽度。 00: 8 位（字节） 01: 16 位（半字） 10: 32 位（字）
DES1	IIN	指定传输数量计数器中内循环计数器的初始值。
	IRM	指定传输数量计数器中内循环计数器的剩余值。
	ORM	指定传输数量计数器中外循环计数器的剩余值。

MODE

MODE 写入 0 时，DSTC 将以模式 0 执行传输。在此模式中，一个启动触发信号便能使 DSTC 传输 TW 所指定位宽的数据 IIN 次。执行 IIN 所指定数量的传输后， DSTC 将执行链路启动或等待下一个启动触发信号。如果向 DSTC 发出 ORM 指定数量的启动触发信号，则 DSTC 将结束 DES 所指定的数量 (IIN×ORM) 的传输。

如果 MODE 写入 1，DSTC 将以模式 1 执行传输。在此模式中，一个启动触发信号便能使 DSTC 传输 TW 所指定位宽的数据一次。执行一次传输后，DSTC 将执行链路启动或等待下一个启动触发信号。如果向 DSTC 发出 IIN×ORM 结果相等数量的启动触发信号，则 DSTC 将结束 DES 所指定的 IIN×ORM 次传输。

SW 传输时可使用模式 0 和模式 1。 HW 传输时，是选模式 0 还是模式 1，具体取决于产生启动触发信号的外设类型。为了使 DSTC 在起始触发时执行传输，来自于外设的起始触发信号已经与 DSTC 传输的每个数据握手，此情况要使用模式 1；若通过 MFS、ADC 和 USB 执行数据传输，因为 DSTC 将数据传输至外设 FIFO，所以使用模式 1；如果是计时器、外部中断时钟等外设的启动触发信号是通知 DSTC 传输启动计时，则模式 0 和模式 1 均可使用。

模式 0 时，指定 ORM 和 IIN 的设置。ORM 和 IIN 中均指定 1 至 65536（含）的一个数。不需指定 IRM 的设置值在 DSTC 中，IIN 设置被复制到 IRM。

模式 1 时，指定 ORM、IIN 和 IRM 的设置。在 ORM 中可指定 1 至 65536（含）中的一个数。在 IIN 中可指定 1 到 256（含）中的一个数。在 IRM 和 IIN 中设置相同值。

3.1.2 传输地址设置

SA、DA、SAC[2:0]、DAC[2:0]

在 DES2 中设置传输源区域 (SA)的启动地址，并在 DES3 中设置传输目标区域 (DA)的启动地址。将各传输地址与指定数据位宽 (TW)对齐。DSTC 无法执行不对齐的传输。

在 DES0 的 SAC[2:0] 位和 DAC[2:0] 位指定传输时更新传输地址的方法。可分别指定为 SA 和 DA 的传输地址更新方法。Table 3-2 列出指定传输地址的方法。

Table 3-2 指定传输地址

区域名称	位	参数
DES0	SAC[2:0] DAC[2:0]	选择传输时更新传输地址 SA 和 DA 的方法。 000: 每次传输时地址按 TW×1 增量，无需内部重载。 001: 每次传输时地址按 TW×1 增量，需要进行内部重载。 010: 每次传输时地址按 TW×2 增量，无需内部重载。 011: 每次传输时地址按 TW×2 增量，需要进行内部重载。 100: 每次传输时地址按 TW×4 增量，无需内部重载。 101: 传输时传输地址保持不变。 110: 每次传输时地址按 TW×1 减量，无需内部重载。 111: 每次传输时地址按 TW×1 减量，需要进行内部重载。
DES2	SA[31:0]	指定传输源区域的启动地址。
DES3	DA[31:0]	指定传输目标区域的启动地址。

更新传输数量计数器和传输地址

Figure 3-1 通过示例说明使用以下设置时如何改变传输数量计数器和传输地址，这些设置包括将外循环传输数量设置成 3，将内循环传输数量设置成 4，以及增加传输地址。本图中的水平轴为时间标度，指示传输进度。

本图上部说明传输数量计数器的行为。内循环计时器剩余值 (IRM) 随每次传输递减计数。已执行 IIN 所指定数量的传输后，IRM 重载内循环计数器初始值 (IIN)。内循环计时器重载时，外循环计时器剩余值 (ORM) 随计时递减计数。DSTC 使用双循环计数器计数传输数量。

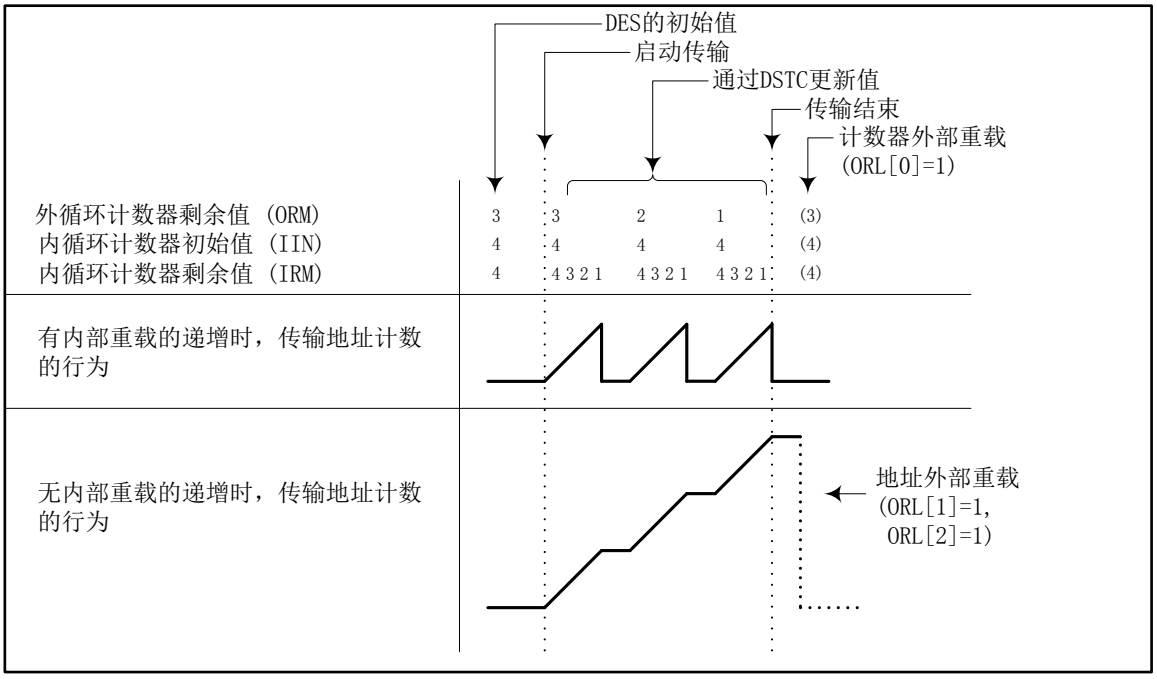
传输时重载 IRM 时，可确定是否执行传输地址 (SA/DA) 恢复至初始值的操作。（内部重载）

本图中间部分所示为传输地址通过内部重载进行递增更新的情况。如图所示，如果启动传输后选择传输地址递增和内部重载，则传输地址增加，且在计时 IRM 重载时，复位至启动传输时的值。

本图下面部分所示为无需内部重载时传输地址递增更新的情况。计时 IRM 重载时，继续递增更新传输地址。

Figure 3-1 也说明外部重载的操作。有关详细说明，参见“外部重载”一节。

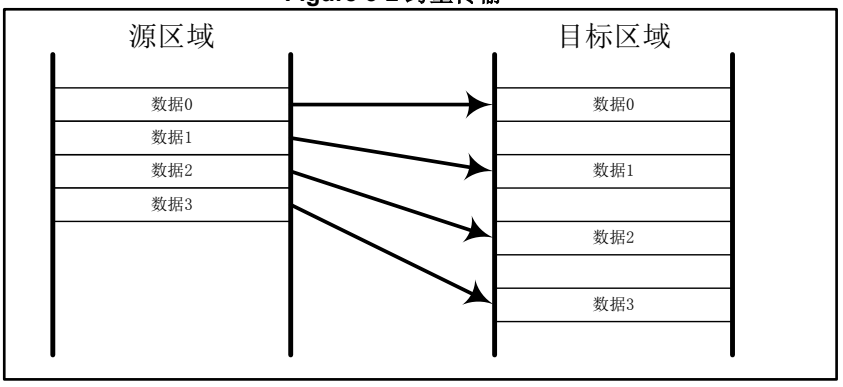
Figure 3-1 传输数量计数器操作和传输地址



跨量传输

如果 SAC[2:0] 位指定 TW×2 递增而 DAC[2:0] 位指定增量 TW×4，则执行跨量传输。每次跨量传输时，传输地址将依次增量 TW×2 和 TW×4。Figure 3-2 所示为 SAC = 000（递增 TW×1）且 DAC = 010（递增 TW×2）时的传输执行示例。同时使用跨量传输和链路传输有助于重组内存数据。有关详细说明，参见 4.3 传输操作示例 3。

Figure 3-2 跨量传输



3.1.3 指定外部重载

ORL[2:0]

Table 3-3 列出指定外部重载传输数量计数器和传输地址的方法。

如 Figure 3-1 所示，IIN×ORM 次传输结束后执行下一个传输时， DES1 中的传输数量计数器 (ORM/IRM/IIN)、DES2 中的传输源地址 (SA)、以及 DES3 中的传输目标地址 (DA) 分别复位（外部重载）为启动传输时的值。

启动传输前，分别将 DES4 设置为 DES1 相同值、 DES5 设置为 DES2 相同值、DES6 设置为 DES3 相同值。IIN×ORM 次传输结束后，DES4、DES5 和 DES6 的值将分别复制到 DES1、DES2 和 DES3，并复位为启动传输前的值。若在下一次传输时使用相同参数，使用外部重载可消除通过 CPU 重建 DES 的必要性。

Table 3-3 指定外部重载

区域名称	位	参数
DES0	ORL[0]	选择是否外部重载传输数量计数器 (DES1) 。 0: 不执行 DES1 外部重载。不需要 DES4 区域。 1: 执行 DES1 外部重载。需要 DES4 区域。
	ORL[1]	选择是否外部重载传输源地址 (DES2) 。 0: 不执行 DES2 外部重载。不需要 DES5 区域。 1: 执行 DES1 外部重载。需要 DES5 区域。
	ORL[2]	选择是否外部重载传输目标地址 (DES3) 。 0: 不执行 DES3 外部重载。不需要 DES6 区域。 1: 执行 DES3 外部重载。需要 DES6 区域。

使用外部重载时 DES 大小

DES0 ~ DES3 是始终需要的区域。只有执行外部重载时才需要 DES4 ~ DES6 区域。如 Table 3-4 所示，根据 ORL[2:0] 位的值确定 DES 的大小和 DES4 ~ DES6 的地址。根据 ORL[2:0] 的值，DES4 ~ DES6 的 DESP 相对地址随 ORL[2:0] 的值而变化。不需要的区域可视为不存在。

Table 3-4 外部重载时 DES 的大小和 DES4 ~ DES7 的储存位置

ORL[2:0]	DES 的大小（字）	DES4 地址	DES5 地址	DES6 地址
000	4	无 DES4	无 DES5	无 DES6
001	5	DESP+0x10	无 DES5	无 DES6
010	5	无 DES4	DESP+0x10	无 DES6
100	5	无 DES4	无 DES5	DESP+0x10
011	6	DESP+0x10	DESP+0x14	无 DES6
101	6	DESP+0x10	无 DES5	DESP+0x14
110	6	无 DES4	DESP+0x10	DESP+0x14
111	7	DESP+0x10	DESP+0x14	DESP+0x18

传输结束后 DES 的值

不使用外部重载功能时，可保存 DES 区域。如果使能内部重载传输地址，就不需要使能外部重装传输地址了，因为启动传输时的值储存在 DES 中。如果传输结束后既未使能外部重载也未使能内部重载，根据

设置的不同，DES 储存的值可能与启动传输时储存的值不同。在这种情况下，由于不能使用相同的 DES 值启动传输，需要通过 CPU 重建 DES。下文说明如何在传输结束后更新 DES 区域的值。

传输结束后 DES0 值与启动传输时的值相同，但 DV[1:0] 位和 ST[1:0] 位的值除外。此外，传输结束后 DES4、DES5 和 DES6 的值与启动传输时的值相同。

如 Table 3-5 所示，传输结束后 DES1 的值（传输数量计数器）根据启动传输时 MODE 和 ORL[0] 的值更新。Table 3-5 中的 "X" 表示操作时该值无效。

传输结束后 IIN 的值与启动传输时的值相同。若 ORL[0] = 0，无论传输启动时的值如何，传输结束后 ORM 值将更新为 0x0001 且 IRM 值更新为 "0x01"。然而，如果启动传输时 ORM 值不为 "0x0001" 或 IRM 值不为 0x01，则 ORM 值或 IRM 值将与启动传输时的值不同。因此，重新启动传输前需要重建 DES，使得 DSTC 通过 CPU 重写 ORM 和 IRM 的必要值。Table 3-5 汇总必要重建 DES1 的条件。

Table 3-5 传输结束时 DES1 的值及重建 DES1 的必要性

传输启动时 DES0/DES1 的值					传输结束后 DES1 的值			重建 DES1 的必要性
MODE	ORL[0]	ORM	IIN	IRM	ORM	IIN	IRM	
0	0	0x0001	X	-	0x0001	值保持	-	不必要
		0x0001 之外的值	X	-				必要
1	0	0x0001	0x01	与 IIN 相同			0x01	不必要
		X	0x01 之外的值	与 IIN 相同				必要
		0x0001 之外的值	X	与 IIN 相同				必要
0	1	X	X	-	复制 DES4 值			不必要
1	1			与 IIN 相同				不必要

如 Table 3-6 所示，传输结束后，根据启动传输时 DES0 中 MODE、SAC[2:0] 和 ORL[1] 的值更新 DES2 值（传输源地址）。Table 3-6 中的 "X" 表示操作时该值无效。如果 DES2 的启动值与结束值不同，重启传输前需重建 DES2。

Table 3-6 传输结束后 DES2 的值

传输启动时 DES0 的值			传输结束后 DES2 的值	重建 DES2 的必要性
MODE	SAC[2:0]	ORL[1]		
0	xx0	0	最终外循环启动时的传输源地址	必要
0	xx1	0	传输启动时的值	不必要
1	xx0	0	最终传输时的传输源地址	必要
1	xx1	0	传输启动时的值	不必要
X	X	1	复制 DES5 值	不必要

如 Table 3-7 所示，传输结束后，根据启动传输时 DES0 中 MODE、DAC[2:0] 和 ORL[2] 的值更新 DES3 值（传输目标地址）。Table 3-7 中的 "X" 表示操作时该值无效。如果 DES3 的启动值与结束值不同，重启传输前重建 DES3。

Table 3-7 传输结束后 DES3 的值

传输启动时 DES0 的值			传输结束后 DES3 的值	重建 DES3 的必要性
MODE	DAC[2:0]	ORL[2]		
0	xx0	0	最终外循环启动时的传输目标地址	必要
0	xx1	0	传输启动时的值	不必要
1	xx0	0	最终传输时的传输目标地址	必要
1	xx1	0	传输启动时的值	不必要
X	X	1	复制 DES6 值	不必要

设置为 DES0.DV[1]=1 且 DES1,2,3 需要重建（DES1,2,3 不恢复为启动值）时，DSTC 将发出 DES 打开错误通知。有关详细说明，参见 3.2.8MONERS 寄存器。

3.1.4 设置链路启动及传输结束通知

CHRS[5:0]、CHLK

收到启动触发信号后，DSTC 按各 DES 所指定次数执行传输（若 MODE = 0，则执行 IIN 次；若 MODE = 1，则执行 1 次)。执行传输后，DSTC 将依据 DES0 中的 CHRS[5:0] 值执行下一进程。Table 3-8 所示为设置链路启动及传输结束通知的方法。

Table 3-8CHRS[5:0]的参数

区域名称	名称	参数
DES0	CHRS[5:4]	这些位选择 DSTC 如何操作 传输数量计数器剩余值变成 (ORM == 1) && (IRM == 1) 后。 00: 不设置中断标志。无链路启动。 DSTC 结束传输。 01: 设置中断标志。无链路启动。 DSTC 结束传输。 10: 不设置中断标志。 DSTC 执行下一个 DES 上的链路启动。 11: 禁止设置（发生 DES 打开错误。）
	CHRS[3:2]	这些位选择 DSTC 如何操作 传输数量计数器剩余值变成 (ORM! = 1) && (IRM == 1) 后。 00: 不设置中断标志。无链路启动。 DSTC 等待启动触发信号。 01: 设置中断标志。无链路启动。 DSTC 等待启动触发信号。 10: 不设置中断标志。 DSTC 执行下一个 DES 上的链路启动。 11: 不设置中断标志。 DSTC 再次执行当前 DES 的链路启动。
	CHRS[1:0]	MODE 为 "1" 时，这些位选择 DSTC 如何操作 传输数量计数器剩余值变成 (IRM! = 1) 后。 00: 不设置中断标志。无链路启动。DSTC 等待启动触发信号。 01: 设置中断标志。无链路启动。 DSTC 等待启动触发信号。 10: 不设置中断标志。 DSTC 执行下一个 DES 上的链路启动。 11: 不设置中断标志。 DSTC 再次执行当前 DES 的链路启动。 MODE 为 "0" 时，上述设置无意义。 MODE 为 "0" 时， CHRS[1:0]写 "00" 。 （MODE 为 "0" 时， CHRS[1:0] 写入 "00" 之外的值将产生 DES 打开错误。
	CHLK	此位选择当前传输后立即执行链路启动的下一个传输（链路锁定）或选择在链路启动的下一个传输前使能执行其他传输。 0: 当前传输后，链路启动传输前执行其他传输。 1: 当前传输后立即执行链路启动传输。

如果下一进程是下一个 DES 上的链路启动，DSTC 将依据下一个 DES 开始传输数据。如果下一进程是当前 DES 的再次链路启动，DSTC 将再次依据当前 DES 开始传输数据。如果下一进程不涉及链路启动，DSTC 将结束传输（或等待下一个启动触发信号）。传输数量计数器的状态确定 DSTC 在当前 DES 后执行下一进程时应遵循 CHRS[5:4]、CHRS[3:2] 还是 CHRS[1:0]。

若不执行链路启动，设置中断标志后，DSTC 可向 CPU 发出通知 DSTC 已结束传输（或正等待下一个启动触发信号)。若为 SW 传输或 SW 传输的链路启动传输，DSTC 将把 SWTR 寄存器的 SWST 位设置成 "1"。若为 HW 传输或 HW 传输的链路启动传输，DSTC 将把 HWINT[n] 寄存器设置成 "1"。

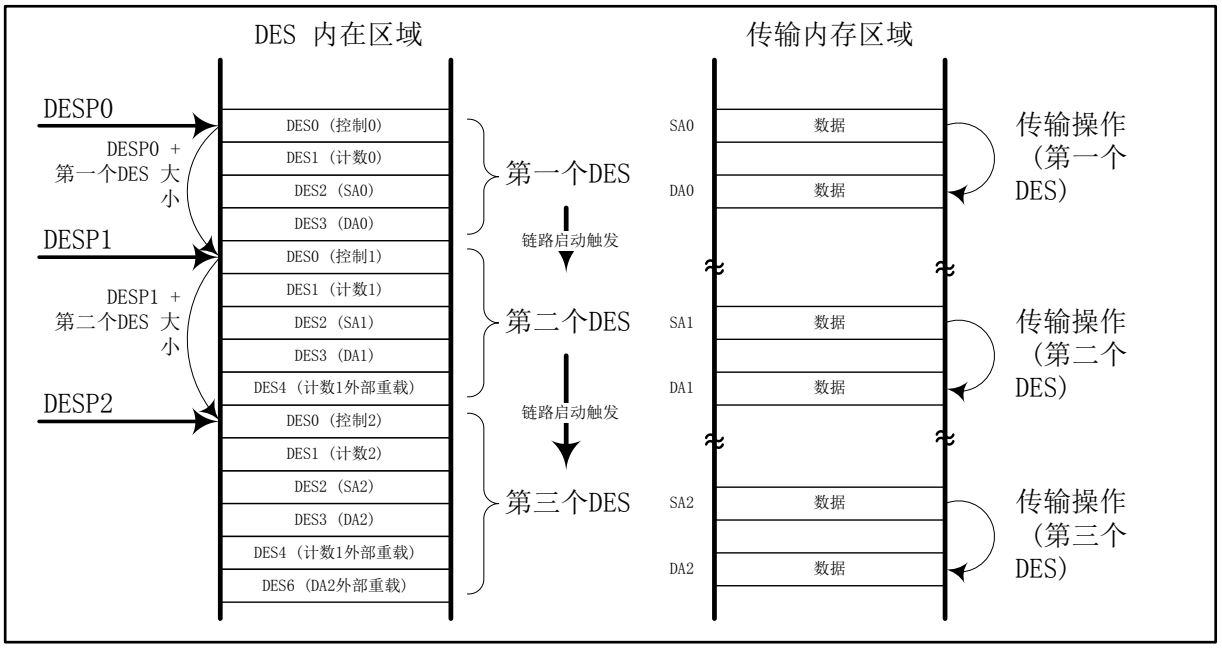
链路启动操作

使用链路启动使能启动触发信号的产生，用于多个 DES 中设置的不同传输。

Figure 3-3 说明 DSTC 执行下一个 DES 上的链路启动时如何执行 DES 引用和传输操作第一个 DES 处于 DESP0 的位置。第一个 DES 的大小通过第一个 DES 的 ORL[2:0] 值定义。如果在第一个 DES 后的 DES 内有链路启动触发信号，DSTC 将通过 DESP0 和第一个 DES 的大小计算后续第二个 DES 的 DESP1。 DSTC 访问第二个 DES 参数后启动第二个 DES 指定的传输。因此，链路启动的后续第二个 DES 必须紧随首先启动的第一个 DES。

指定第二个 DES 中的链路启动可以启动第三个 DES 的传输。DSTC 可保持执行链路启动传输，直到在可设置的最大数量 DES 内再无其他链路启动触发信号。

Figure 3-3 链路启动传输中的 DES 引用



链路锁定功能

如果在链路启动时序有其他源发出的传输启动请求，DSTC 可依据传输优先顺序首先执行其他传输。这时，DSTC 将在执行其他传输后执行指定的链路启动传输。链路锁定功能可确保出现上述情况下，无论传输优先顺序如何，将会连续执行链路启动的传输且其他传输请求不能中断传输。在 DES 中的 CHLK 写 1，可使能链路锁定功能。

3.1.5 其他 DES 设置

DV[1:0]

执行传输操作时，DSTC 将访问并更新 DES 区域。如果 CPU 更新 DSTC 正使用的 DES 区域，DSTC 可能执行程序未设置的非法传输操作。 DES 区域 CPU 和 DSTC 之间使用互相排斥的内存管理体系，防止非法传输操作。在 DES0 中，DV[1:0] 位（描述符有效）指定 DES 写更新所有权属于 CPU 或 DSTC。Table 3-9 列出 DV 的参数和 DSTC 相关操作。

Table 3-9DV 的参数

区域名称	名称	参数
DES0	DV[1:0]	DV 指定 DES 所有权属于 CPU 还是 DSTC。 DV 指定是否在 DES 打开进程后执行传输。 DV 指定是否在传输结束后执行 DES 关闭进程。 00: 所有权属于 CPU。不执行传输。不执行 DES 关闭进程。 （DSTC 读取此值时，DSTC 将通知出现 DES 打开错误。） 01: 所有权属于 DSTC。执行传输。执行 DES 关闭进程。 10: 所有权属于 DSTC。不执行传输。执行 DES 关闭进程。 11: 所有权属于 DSTC。执行传输。不执行 DES 关闭进程。

如果 DV 为 00，表示 DES 区域的所有权属于 CPU。若 DV 为 01、10 或 11，表示 DES 区域的所有权属于 DSTC。设置 DES 初始值后，CPU 设置 DV 为 01、10 或 11，通知该 DES 的所有权属于 DSTC。收到启动触发信号后，DSTC 检验 DES0 的 DV 值，确定后续操作。（DSTC 根据启动触发信号读取 DES 的进程称为 DES 打开进程。）

若 DV 设为 01，DSTC 将按指定次数 (ORM×IIN) 执行传输。传输正在进行时，DV 值保持为 01 且 DSTC 保持对 DES 的所有权。所指定次数的所有传输完成时，DSTV 将 DV 值更新为 00，DSTC 将 DES 的所有权归还 CPU。（DSTC 将 DV 值更新为 00 且 DSTC 将 DES 的所有权归还 CPU 的进程称为 DES 关闭进程。）

若 DV 被设置成 11，DSTC 将按指定次数 (ORM×IIN) 执行传输。传输正在进行时，DV 值保持为 11 且 DSTC 保持对 DES 的所有权。即使已所指定次数的所有传输已经完成，DSTC 也不会执行 DES 关闭进程并继续保留 DES 的所有权。

若 DV 被设置成 10，DSTC 将不执行传输而是仅执行 DES 关闭进程。

若 DV 被设置成 00，DSTC 识别 CPU 正在更新 DES 区域，不执行任何传输或更新 DES。DSTC 通知 CPU 出现 DES 打开错误。

程序按以下所述 DES 的使用方法确定 DV 值。

如果每次传输时 DES 所定义的传输参数会有变化，且 CPU 必须更新 DES 参数，则将 DV 设为 01，并将 DES 的所有权赋予 DSTC。在这种情况下，检验 DV 值为 00 且 DES 的所有权已经归还 CPU 后，CPU 可安全更新 DES。完成 DES 初始化后，CPU 再次将 DES 的所有权移交给 DSTC。

如果 DES 定义的传输参数为固定参数且 CPU 必须重复使用 DES 参数，则将 DV 设置成 "11"，并将 DES 的所有权赋予 DSTC。在这种情况下，由于传输后没有执行 DES 关闭进程，可省略 CPU 再次移交 DES 所有权的进程。指定次数 (ORM×IIN) 的传输完成后，若发出新的启动触发信号，则启动和先前参数相同的传输。

DV 被设置成 11 时，由于 DSTC 保持对 DES 的所有权，未归还 CPU，传输将变成无限循环进程。要避免传输的无限循环，将 DV 设置成 "10"，将 DV 设为 "11" 时移交给 DSTC 的 DES 所有权归还 CPU。

若 DV 设为 00，CPU 可更新 DES。若 DV 设为 01、10 或 11，DSTC 根据 DES 执行传输时，CPU 不能更新除 DV 之外的 DES 区域参数。

DV[1] 设为 1 时，DSTC 将在执行 ORM×IIN 次传输后重复使用 DES 值。因此，增加重载传输计数器和传输地址的具体限制条件。如果 DV[1] 设为 1 且 DES 的设置使得 DES1、DES2 和 DES3 值无法恢复相应的值，则 DSTC 通知 CPU 出现 DES 打开错误。有关详细说明，参见 “3.2.8MONERS 寄存器”。

ST[1:0]

DES0 的 ST(状态)[1:0]是 DSTC 用于将传输结束状态通知 CPU 。Table 3-10 列出 DSTC 在 DES 关闭进程中写入 ST 的数据。

Table 3-10ST 通知的内容

区域名称	名称	参数
DES0	ST[1:0]	DES 指定的传输结束后，DSTC 在 DES 关闭进程中将结束状态值写入 ST。 00: 传输正常结束。 01: 传输源访问出现错误，传输异常结束。 10: 传输目标访问出现错误，传输异常结束。 11: CPU 发出的待机转换命令强制性停止传输，传输异常结束。

如果由于 DES 访问错误或 DES 打开错误异常结束传输，则 DSTC 不执行 DES 关闭进程，也不将数据写入 ST，因为 DSTC 无法读取 DES 区域。此类错误通知根据 DSTC 的 MONERS 寄存器执行，不是 DES 的 ST。有关详细说明，参见 3.2.8 MONERS 寄存器。

PCHK[3:0]

PCHK[3:0]（奇偶校验）设置 DES0 区域的奇偶校验（调用以下方程式）。

PCHK[3:0] != (DES0[27:24] ^ DES0[23:20] ^ DES0[19:16] ^ DES0[15:12] ^ DES0[11:8] ^ DES0[7:4])

如果 CPU 失控等事件破坏 DES 区域内的数据，DSTC 可能会启动意外的传输。DES0 内配置奇偶校验功能，防止发生上述事件。首次构建 DES 时，CPU 在 PCHK 设置 DES0 的奇偶校验。 DSTC 在 DES 打开进程中检查 PCHK 值和 DES0 值的一致性。如果出现奇偶校验错误，DSTC 将通知 CPU 出现 DES 打开错误且不执行传输。有关详细说明，参见 3.2.8 MONERS 寄存器。

ACK[1:0]

ACK[1:0]（确认）设置的值用于调节使用 HW 传输时 DSTC 输出 DMA 传输请求确认信号至外围设备的时序。

如果使用 HW 传输，将外围设备的 HW 启动直接启动的 DES 的 ACK 设置成 01。若为其他 DES（HW 传输链路启动的 DES、SW 传输使用的 DES、以及 SW 传输链路启动的 DES），则将 ACK 设置成 00。

DMSET

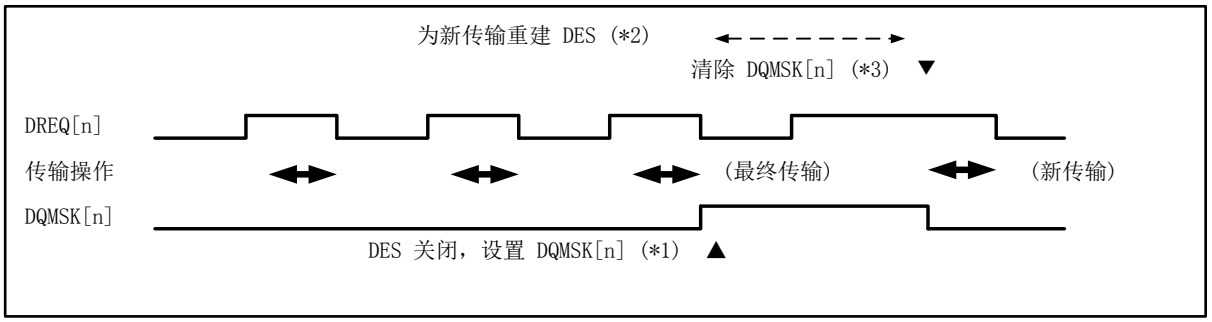
在 DSTC 完成 DES 关闭进程与 CPU 完成 DES 重建之间的周期内，DMSET（DMA 请求屏蔽设置）的功能是设置 DQMSK[n] 寄存器以及屏蔽外设发出的 DMA 传输请求信号。有关详细说明，参见 “3.2.4 HW 传输的控制”。

如果使用 HW 传输，传输结束后，DMA 传输请求确认信号 (DACK) 使外设的 DMA 传输请求信号 (DREQ)无效。然而，具体取决于外设的类型，DREQ 在以下传输请求时生效，无论 DSTC 的状态如何。如果 DSTC 完成 DES 关闭进程与下一个传输启动就绪（DES 重建）之间的周期内 DREQ 生效，因下一个传输启动未就绪，DSTC 通知 CPU 出现 DES 打开错误。在这种情况下，设置描述符 0 至 1 内的 DMSET 位可防止 DSTC 通知 CPU 出现 DES 打开错误，还可抑制 HW 启动传输，直到完成 DES 重建才启动。

Figure 3-4 所示为操作示例。

如果 DES0 内 DMSET = 1 是由外设的 HW 传输直接处理，或通过 HW 传输的链路启动方式启动，执行 DES 关闭进程时，DQMSK[n] 寄存器中 HW 通道对应的位设为 1。（*1，如 Figure 3-4 所示）该通道位设为 1 后，DSTC 不能识别 DREQ[n] 信号，也不通知 CPU 有 DES 打开错误。CPU 重建 DES（*2，如 Figure 3-4 所示）且下一个传输准备就绪后，CPU 清除 DQMSK[n] 寄存器中的通道位。Figure 3-4（*3，如 Figure 3-4 所示）CPU 清除 DQMSK[n] 寄存器中的通道位后，DSTC 将识别 DREQ[n] 信号。根据 CPU 重建的 DES 启动新的传输。

Figure 3-4 使用 DMSET 抑制传输启动



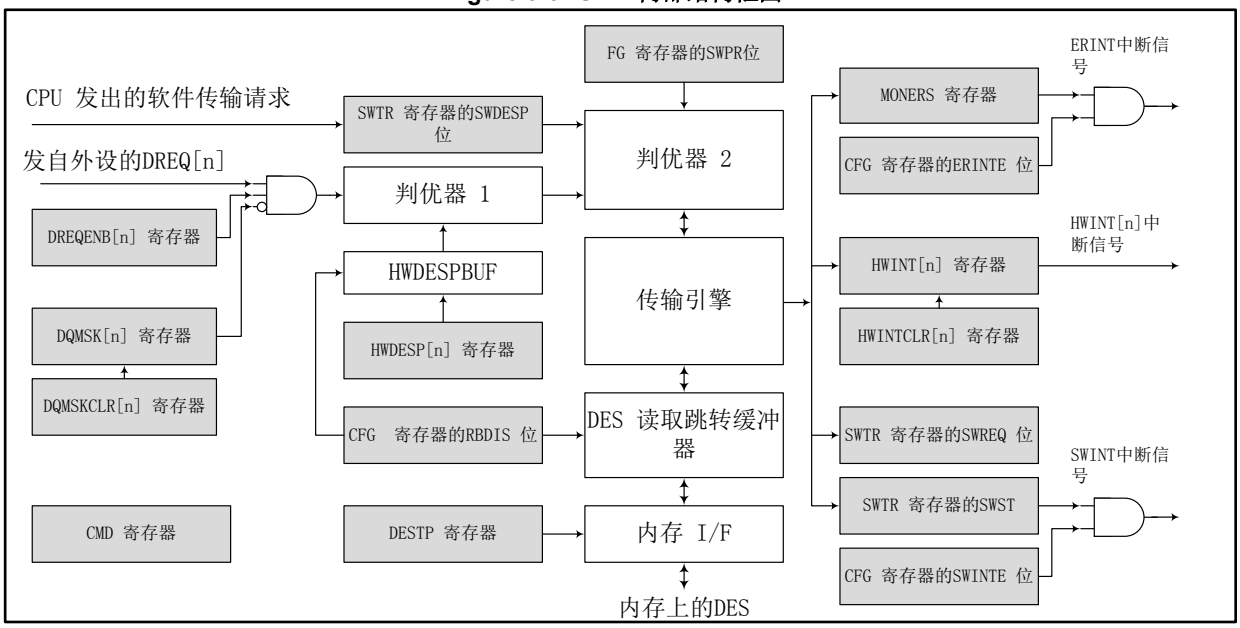
3.2 DSTC 的控制功能

本节说明 DSTC 的控制功能。

3.2.1 DSTC 内部结构框图

Figure 3-5 说明 DSTC 内可从 CPU 访问的控制模块和控制寄存器（阴影方框）之间的连接。CPU 通过访问控制寄存器启动 DSTC 传输并控制结束通知。以下章节说明各模块的操作并概述各寄存器功能，如 DSTC 内部结构框图所示。有关寄存器功能的详细说明，参见 “5DSTC 寄存器和描述符”。

Figure 3-5DSTC 内部结构框图



3.2.2 DESTP 寄存器

DESTP（DES 顶部地址）寄存器指定内存上 DES 区域的启动地址。进行初始设置时指定启动地址。DSTC 访问 "DESTP + DESP" 地址的 DES 并执行传输。

3.2.3 SW 传输控制

发出 SW 传输启动触发信号时，将要启动的 DES 的 DESP 值写入 SWTR（软件触发信号）寄存器中的 SWDESP（软件 DES 指针）。如果在 SW 传输进程中执行链路启动，DSTC 将更新 SWDESP 为链路启动所使用的 DESP 值。SWDESP 值被作为传输请求发送至 Figure 3-5 所示的判优器 2。

SWTR 寄存器中的 SWREQ（软件请求）位为只读位，指示 SW 传输是挂起的还是正在执行 SW 传输和链路启动传输。写入（启动触发） SWTR 寄存器，将 SWREQ 位设置为 1。如果 SW 传输正常结束、异常结束或正等待启动触发信号，SWREQ 复位为 0。

只有在当前 SW 传输结束后才能发出 SW 启动触发信号。SWREQ 位为 1 时，忽略对 SWTR 寄存器的写入。

SWTR 寄存器中的 SWST（软件状态）位为只读位，用于将 SW 传输结束通知发送至 CPU。在 SW 传输 DES 内的 CHRS 中或在通过 SW 传输链路启动的 DES 中指定中断标志设置。如果 SW 传输正常结束，SWST 设为 1。可向 CMD 寄存器发出 SWCLR 命令，将 SWST 清除为 0。

如果 SWST 位已设为 1，CFG 寄存器的 SWINTE 位写入 1 可使能 SWINT 中断。若 (SWINTE==1)&(SWST==1)，NVIC 的 SWINT 中断信号生效。

3.2.4 HW 传输控制

如果外设发出传输请求（DREQ[n] 生效），则 DSTC 启动 HW 传输。DSTC 使用以下寄存器控制传输通道上的 HW 传输，寄存器编号对应传输通道编号。外设发出传输请求前，CPU 对寄存器进行初始设置。此外，CPU 根据传输进度清除寄存器。

DREQENB[n] 寄存器

DREQENB[n]（DMA 请求使能）寄存器确定是否在初始设置中使用 HW 通道 n。 DREQENB[n] 寄存器写 1，使用 HW 通道 n。DREQENB[n] 寄存器写 0，不使用 HW 通道 n。DREQENB[n] 寄存器为 0 时，与 DSTC 连接的外设中断信号 (DREQ[n]) 被忽略。 DSTC 不修改 DREQENB[n] 寄存器的值。

DREQENB[n] 寄存器值确定选择外设中断信号还是 DSTC 的 HWINT[n]，用作连接 NVIC 的中断信号。有关详细说明，参见 “2 DSTC 操作概述及 DSTC 系统配置”。

DQMSK[n] 寄存器和 DQMSKCLR[n] 寄存器

DQMSK[n]（DMA 请求屏蔽) 寄存器为只读寄存器。本寄存器为 1 时表示发送至 DSTC 的 HW 启动请求 (DREQ[n]) 受到抑制。满足以下任意条件时，DSTC 设 DQMSK[n] 为 1 并抑制对应 DQMSK[n] 的 HW 通道传输请求。

- HW 通道 n 的 HW 传输时发生传输错误。
- CPU 已向 CMD 寄存器发出待机转换命令。
- HW 通道 n 上用于传输的 DES 中 DMSET 为 1，且 DSTC 已执行 DES 关闭进程。

CPU 已重建 DES 且 HW 传输启动已准备就绪后，CPU 可解除抑制发出给 DSTC 的 HW 启动传输请求。如果 DQMSKCLR[n]（DMA 请求屏蔽清除）寄存器写入 1，DQMSK[n] 寄存器清除为 0 并识别随后的 HW 传输请求 (DREQ[n])。

HWDESP[n] 寄存器

HWDESP[n]（硬件 DES 指针）寄存器设置执行 HW 通道 n 的传输请求时 DSTC 访问和执行的 DES 的 DESP。发出 HW 传输请求前设置本寄存器。

发出 HW 启动触发信号时，DSTC 将参考 HWDESP[n] 寄存器内所设置 DES 的 DESP 启动传输。如 Figure 3-5 所示，DSTC 在使用前将 HWDESP[n] 寄存器的 DESP 值储存在 HWDESPBUF 中。链路启动时， HWDESPBUF 中储存在的值更新为链路启动后设置的 DESP 值。 DSTC 不能修改 HWDESP[n] 寄存器值。

如果相继发出通道 n 的 HW 启动请求，DSTC 使用 HWDESPBUF 中储存的 DESP 值而不是 HWDESP[n] 寄存器的 DESP 值。因此，通过 CPU 修改 HWDESP[n] 寄存器值时， HWDESPBUF 中储存的值无效。修改 CFG 寄存器中的 RBDIS 位值可使 HWDESPBUF 的 DESP 值无效。有关详细说明，参见 5.5 CFG 寄存器。

HWINT[n] 寄存器和 HWINTCLR[n] 寄存器

HWINT[n]（硬件传输中断）寄存器为只读寄存器，用于将 HW 传输结束通知发送至 CPU。在 HW 启动 DES 后，在 HW 启动的 DES 或链路启动的 DES 中的 CHRS 内指定中断标志设置。如果 HW 传输正常结束，HWINT[n] 设为 1。HWINTCLR[n] 寄存器写入 1 可将 HWINT[n] 寄存器清除为 0。HWINT[n] 寄存器设为 1 时，NVIC 的中断信号生效。

3.2.5 传输请求的判优

如果多个 HW 启动请求与 SW 启动请求冲突，DSTC 将判定启动触发信号的优先性并按顺序执行传输。如 Figure 3-5 所示，启动请求判优由判优器 1 和判优器 2 这两个模块处理。判优详细说明如下。

判优器 1

判优器 1 对 HW 传输请求进行判优。如果请求有冲突，判优器 1 将使用下文所述循环法选择传输启动通道。总线复位后，通道编号越小，在优先顺序选择中的优先级越高。

最高优先级 0,1,2,3,4,5,6,7,,,,,,254,255 最低优先级

比如，根据此优先顺序，若通道 5 和通道 6 同时发出请求，选择通道 5。一旦被选为传输通道后，其优先级将轮换为最低优先级。在上述示例中，由于选择了通道 5，优先顺序将更新如下：

最高优先级 6,7,8,9,10,11,,,,,,254,255, 0,1,2,3,4,5 最低优先级

比如，根据此优先顺序，若通道 5 和通道 6 同时发出请求，选择通道 6。循环法使多个 HW 传输请求能得到平等处理。

判优器 1 访问所选通道 n 的 HWDESP[n] 并通知判优器 2 所用 DES 的 DESP。此外，如果 HW 传输使用链路启动，则判优器 1 将通知判优器 2 更新的 DESP。所有链路传输已结束且 DSTC 已启动等待下一个启动触发信号后，判优器 1 通知判优器 2 所选通道 n 的传输请求。

判优器 2

判优器 2 选择是执行判优器 1 选择的 HW 传输请求还是执行 SW 传输请求。如果请求有冲突，DSTC 将根据 CFG 寄存器的 SWPR（软件传输优先性）位所设置概率集选择传输请求，并启动传输引擎。Table 3-11 列出 CFG 寄存器 SWPR 位的设置和 SW 传输获得传输权的概率。

Table 3-11 CFG:SWPR[2:0]的参数

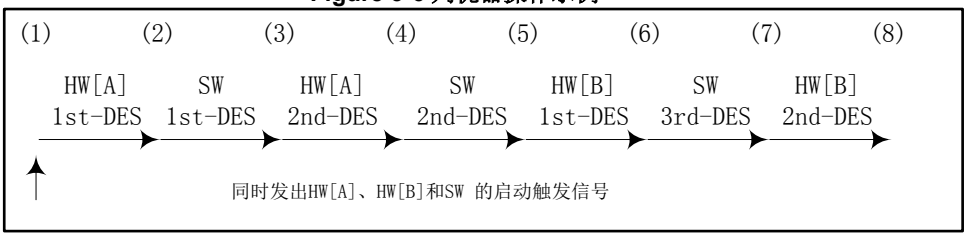
区域名称	名称	参数
CFG	SWPR[2:0]	在判优器 2 判优时，如果 SW 传输请求与 HW 传输请求冲突， 判优器 2 指定 SW 传输获得传输权的概率。 000: 将 SW 传输的优先级设置成最高优先级。（如果在 HW 传输正在进行时发出 SW 传输请求，将在 HW 传输结束后启动 SW 传输。） 001: 将 SW 传输获得传输权利的概率设置成 1/2。 010: 将 SW 传输获得传输权利的概率设置成 1/3。 011: 将 SW 传输获得传输权利的概率设置成 1/7。 100: 将 SW 传输获得传输权利的概率设置成 1/15。（初始值） 101: 将 SW 传输获得传输权利的概率设置成 1/31。 110: 将 SW 传输获得传输权利的概率设置成 1/63。 111: 将 SW 传输的优先级设置成最低优先级。（只有在无 HW 传输请求时才能启动 SW 传输。）

判优器操作示例

Figure 3-6 所示为操作示例。此图中的水平轴为时间轴。本图说明 DSTC 判优进程中不同传输请求获得传输权利的顺序。传输源有三个：HW 通道 A 传输，HW 通道 B 传输和 SW 传输。HW[A] 传输和 HW[B] 传输通过两个 DES 连接为链路传输；SW 传输通过三个 DES 在连接为链路传输。CFG 寄存器中的 SWPR 位设为 001（SW 传输的概率为：1/2）。没有在任何 DES 中指定链路锁定。

在时序 (1) 中，同时发出 HW[A] 传输、HW[B] 传输和 SW 传输的传输请求。判优器 1 对冲突的 HW[A] 传输和 HW[B] 传输进行判优。判优器 1 根据先前的循环状态选择 HW[A] 传输或 HW[B] 传输。下文假定判优器 1 已选择 HW[A] 传输。判优器 2 对冲突的 HW[A] 传输和 SW 传输进行判优。判优器 2 根据先前的循环状态选择 HW[A] 传输或 SW 传输。下文假定判优器 2 已选择 HW[A] 传输。DSTC 的传输引擎启动 HW[A] 的第一个 DES 传输。

Figure 3-6 判优器操作示例



在时序 (2) 中，HW[A]1st-DES 传输结束。发出 HW[A]2nd-DES 的链路启动请求。（保持 HW[B] 请求，直到再无 HW[A] 链路启动为止。）判优器 1 请求判优器 2 选择 HW[A] 的第二个 DESP。判优器 2 对冲突的 HW[A] 传输和 SW 传输进行判优。由于 SW 传输的概率为 1/2 且判优器 2 已在时序 (1) 中选择 HW[A] 传输，判优器 2 将选择 SW1st-DES 传输。

在时序 (3) 中，SW1st-DES 传输结束且发出 SW2nd-DES 链路启动请求。判优器 2 对冲突的 HW[A]2nd-DES 传输和 SW-2ndDES 传输进行判优。由于 SW 传输的概率为 1/2 且判优器 2 已在时序 (2) 中选择 SW 传输，判优器 2 将选择 HW[A]2nd-DES 传输。

在时序 (4) 中，HW[A]-2ndDES 传输结束。判优器 1 向判优器 2 发出传输 HW[B]1st-DES 的请求。判优器 2 对冲突的 HW[B]1st-DES 传输和 SW2nd-DES 传输进行判优，然后选择 SW2nd-DES 传输。

在时序 (5)、(6) 和 (7) 中，判优器 2 执行相同的判优操作并选择 HW[B]2nd-DES 传输和 SW3rd-DES 传输。

如上文所述，可在 HW 传输链路传输进程中执行 SW 传输，可在 SW 传输链路传输进程中执行 HW 传输。在 HW 传输链路传输进程中，不执行其他任何通道的 HW 传输。如果 DES 中已指定链路锁定，无论 CFG 寄存器的 SWPR 位如何设置，已执行该 DES 传输后，始终相继执行链路启动中的 DES 传输。

在上述示例中，由于在 CFG 寄存器 SWPR 位中设置的概率为 1/2，每两次传输执行一次 SW 传输。考虑同时使用的 DSTC 的 HW 传输通道数量、DES 中链路数量、各传输中传输数据量等，将 CFG 寄存器 SWPR 位设置成适当值。即使 DSTC 执行传输时也可以修改 CFG 寄存器 SWPR 位的值。修改 CFG 寄存器 SWPR 位的值后，可应用于下一个 SW 启动触发。

3.2.6 读取跳转缓冲功能

执行传输时，DSTC 传输引擎访问内存中 DES 的传输信息。如果所有传输不是在一个启动触发信号情况下结束，DSTC 将已执行 DES 的传输数量和传输地址反写至 DES。如果传输地址固定或不需要更新，DSTC 将跳过反写进程。在下一次启动触发时，DSTC 继续根据更新的 DES 信息执行传输。

由于 DSTC 在每次启动触发时要花时间访问 DES，DSTC 具有如 Figure 3-5 所示的读取跳转缓冲功能。DSTC 将 DSTC 已读取的 DES 传输信息储存在内部读取跳转缓冲器内。下一个启动触发信号访问与当前启动触发信号相同的 DESP 时，DSTC 将不访问内存中的 DES 而是使用读取跳转缓冲器中的值执行传输，加快处理速度。

使用 CFG 寄存器中的 RBDIS 位可使能和禁用读取跳转缓冲功能。实际应用时，使能读取跳转缓冲功能可加快传输速度。CFG 寄存器的 RBDIS 位写 1 禁用读取跳转缓冲功能时，始终直接访问内存的 DES。

若通过 CPU 修改 DES0 内 DV 的值为 "10" 停止 DSTC 的传输操作（退出无限循环），修改 DES0 内 DV 的值后， CFG 寄存器的 RBDIS 位写入 "1" 可使读取跳转缓冲器中储存的 DES 信息无效。只要保持使能读取跳转缓冲功能，DSTC 可能会跳过对 DES 的访问，而不能识别到通过 CPU 改变的 DV 值。有关详细说明，参见 “5.5 CFG 寄存器”。

3.2.7 传输结束操作

如果 DSTC 正常结束传输，则根据 DES 参数执行链路启动，设置中断标志（SWTR:SWST 或 HWINT[n]）和 DES 关闭进程。

如果发生传输错误时，传输将立即中断。这种传输中断称为错误结束。错误结束时，DSTC 不执行链路启动。此外，DSTC 不设置中断标志（SWTR:SWST 或 HWINT[n]），而是将发生错误的参数记录在 MONERS 寄存器中。错误的参数确定 DSTC 是否执行 DES 关闭进程。

HW 传输时，如果发生传输错误，DSTC 设置发生传输错误的通道所对应的 DQMSK[n] 寄存器，并抑制该通道之后的 HW 传输请求。

由于发生错误的 DES 参数仍然与传输进程中的参数相同，启动新传输前，应通过 CPU 重建 DES 区域。

3.2.8 MONERS 寄存器

如果发生传输错误，错误的参数将被记录在 MONERS 寄存器中。Table 3-12 列出 MONERS 寄存器显示的参数。

Table 3-12 MONERS 的参数

区域名称	名称	参数
MONERS	EST[2:0]	指示已发生错误的参数。 000: 未发生错误。 001: 源访问错误 010: 目标访问错误 011: 待机转换命令产生的传输强制停止错误 100: DES 访问错误 101: DES 打开错误 上述值以外的值：未定义
	DER	此位指示是否发生双重错误。 0: 指示未发生双重错误。 1: 指示已发生双重错误。
	ESTOP	指示 DSTC 是否处于错误停止状态。 0: 指示 DSTC 未处于错误停止状态。 1: 指示 DSTC 处于错误停止状态。
	EHS	指示 HW 启动或 SW 启动是否已启动造成错误的 DES。 0: SW 启动或该 SW 启动的链路启动所启动的传输已发生错误。 1: HW 启动或该 HW 启动的链路启动所启动的传输已发生错误。
MONERS	ECH	如果 HW 启动已启动造成错误的 DES，指示 HW 通道编号。
	EDESP	指示已产生错误的 DES 的 DESP。

访问 MONERS 寄存器可以检验发生传输错误的参数。MONERS 寄存器中的 EST[2:0]位指示已发生错误，CFG 寄存器的 ERINTE 位写入 1 可使能 ERINT 中断。 ERINTE 位设为 1 时，NVIC 的 ERINT 中断信号生效。向 CMD 寄存器发出 ERCLR 命令可清除 MONERS 寄存器的值和 ERINT 中断。可能发生错误的参数说明如下：

DES 访问错误

如果 DSTC 访问 DESTP+DESP 的 DES 区域时发生以下任意事件，DSTC 将以错误结束（DES 访问错误）形式结束传输。DSTC 将 MONERS:EST 设为 100。发生 DES 访问错误时，DSTC 不执行 DES 关闭进程。

- 计算的 DES 区域地址值溢出（在 0x00000000-0xFFFFFFFF 范围之外）。
- 访问 DES 区域时，DSTC 收到系统的总线错误响应。

DES 打开错误

DSTC 已访问 DESTP+DESP 区域后，如果 DES0 或 DES1 值满足以下任意条件，则 DSTC 将其视为异常 DES 指定值并以错误结束（DES 打开错误）形式结束传输。DSTC 将 MONERS:EST 设为 "101"。发生 DES 打开错误时，DSTC 不执行 DES 关闭进程。

- DV[1:0]==00（无 DES 所有权）
- PCHK[3:0] != (DES0[27:24] ^ DES0[23:20] ^ DES0[19:16] ^ DES0[15:12] ^ DES0[11:8] ^ DES0[7:4])（DES0 奇偶校验错误）
- DES0 保留区两位中任一位为 1。（异常指定值）
- TW[1:0]==11（异常指定值）
- CHRS[5:4]==11（异常指定值）
- (CHRS[5]==0) &&(CHRS[3]==0)&&(CHRS[1]==0) &&(CHLK ==1)（异常链路设置）
- (MODE==0) && (CHRS[1:0] != 00)（异常设置）
- (MODE==0)&&(ORM==0x0000) && (IIN≥0x2000)（超出模式 0 允许的计数值范围）
- (MODE==0)&&(ORM≥0x8000) && (IIN≥0x4000)（超出模式 0 允许的计数值范围）
- (MODE==0)&&(ORM≥0x4000) && (IIN≥0x8000)（超出模式 0 允许的计数值范围）
- (MODE==0)&&(ORM≥0x2000) &&(IIN==0x0000)（超出模式 0 中允许的计数值范围）
- (MODE==1)&&(IIN!=0x00)&&(IRM==0x00)（超出模式 1 允许的计数值范围）
- (MODE==1)&&(IIN!=0x00)&&(IRM□IN)（超出模式 1 允许的计数值范围）
- (MODE==0)&&(DV[1]==1)&&(ORL[0]==0)&& (ORM != 0x0001)（异常计数器重载设置）
- (MODE==1)&&(DV[1]==1)&&(ORL[0]==0)&& (ORM != 0x0001)（异常计数器重载设置）
- (MODE==1)&&(DV[1]==1)&&(ORL[0]==0) && (IRM != 0x01)（异常计数器重载设置）
- (MODE==1)&&(DV[1]==1)&&(ORL[0]==0)&& (IIN != 0x01)（异常计数器重载设置）
- (DV[1]==1)&&(SAC[0]==0)&&(ORL[1]==0)（异常传输源地址重载设置）
- (DV[1]==1)&&(DAC[0]==0)&&(ORL[2]==0)（异常传输目标地址重载设置）

源访问错误

如果 DSTC 访问传输源地址区域时发生以下任意事件，DSTC 将以错误结束（源访问错误）形式结束传输。DSTC 将 MONERS:EST 设为 001。同时 DSTC 在 DES0:ST 写入 01 并执行 DES 关闭进程。

- 指定传输源启动地址值 (SA) 与 TW 不匹配。
- 传输源地址值有递增或递减计算溢出。
- DSTC 收到系统的总线错误响应。

目标访问错误

如果 DSTC 访问传输目标地址区域时发生以下任意事件，DSTC 将以错误结束（目标访问错误）形式结束传输。DSTC 将 MONERS:EST 设为 010。同时 DSTC 在 DES0:ST 写入 10 并执行 DES 关闭进程。

- 指定传输目标启动地址值 (DA) 与 TW 不匹配。
- 传输源地址值有递增或递减计算溢出。
- DSTC 收到系统的总线错误响应。

传输强制停止错误

执行传输时，如果 DSTC 收到 CPU 的待机转换命令，将结束传输（传输强制停止错误）。DSTC 将 MONERS:EST 设为 011。同时 DSTC 将 11 写入 DES0:ST 并执行 DES 关闭进程。

DER 功能和 ESTOP 功能

如果发生传输错误，引起传输错误的 DES 传输被中断，并结束。传输结束后，如果有另一个 DES 的传输启动请求，寄存器中 ESTE（错误停止使能）位的设置，确定 DSTC 是否启动传输启动请求的传输。

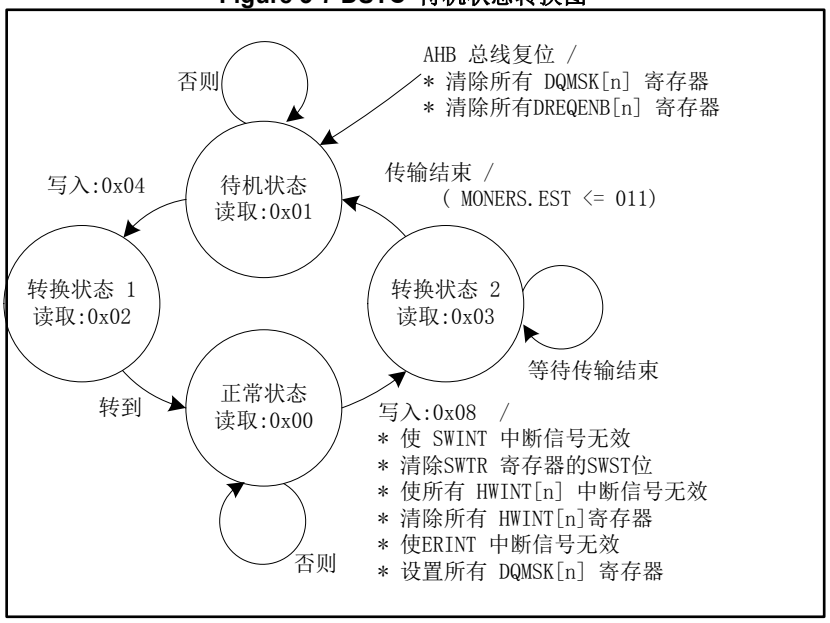
CFG:ESTE = 0 时，如果发生传输错误后有新的传输请求，DSTC 启动新传输请求的传输。MONERS 寄存器记录并保持错误信息，直到 CPU 清除寄存器。 MONERS 寄存器保持错误信息 (EST ≠ 000) 时，如果另一传输请求引起的传输因错误结束，DSTC 将 DER（双重错误）位设为 1。DER 位指示发生了双重错误。对于第二个错误，DSTC 只通知 CPU 发生了此错误。MONERS 寄存器只保留第一个错误的信息，不保留第二个错误的参数。此外，MONERS 寄存器也不记录在第二个错误之后的任何错误。

CFG:ESTE 设为 1 时，如果发生传输错误，DSTC 将转换至错误停止状态。转换至错误停止状态后，DSTC 保留其他传输请求不再启动任何传输。将 MONERS 寄存器中的 ESTOP 位设为 1 指示 DSTC 处于错误停止状态。如果 CPU 向 CMD 寄存器发出 ERCLR 命令，DSTC 将解除错误停止状态并根据其保留的传输请求启动传输。

3.2.9 待机功能

DSTC 具有停止 DSTC 内部时钟的功能（待机功能），使 DSTC 停止操作，降低功耗。DSTC 的状态可以通过发出待机转换命令和待机释放命令到 CMD 寄存器进行切换。Figure 3-7 说明 DSTC 在发出待机转换命令并处于状态转换时执行的操作。

Figure 3-7 DSTC 待机状态转换图



DSTC 具有四种状态：待机状态、转换状态 1、正常状态和转换状态 2。可通过 CPU 读取 CMD 寄存器值检查 DSTC 的状态。

总线复位后，DSTC 的初始状态为待机状态。CPU 向 CMD 寄存器发出待机释放命令（写入 0x04）时，DSTC 将转换至转换状态 1 ，然后转换至正常状态。

正常状态时，如果 CPU 向 CMD 寄存器发出待机转换命令（写入 0x08），DSTC 将转换至转换状态 2，等待转换结束。

转换状态 2 时，如果 DSTC 未执行任何传输，将立即转换至待机状态。但是，如果 DSTC 执行传输，则在传输被强制结束后转换至待机状态。

如果通过发出待机转换命令结束传输，DSTC 将 11 （强制结束代码）写入该传输 DES 的 ST 中，执行 DES 关闭进程。此外，MONERS 寄存器中的 EST 位设为 011。DSTC 收到 HW 传输和 SW 传输时，执行两种传输的 DES 关闭进程。

此外，CPU 向 DSTC 发出待机转换命令时，DSTC 将同时执行以下进程：

- DSTC 使 SWINT 中断信号无效并清除 SWTR:SWST。
- DSTC 清除所有 HWINT[n] 寄存器并使所有 HWINT[n] 中断信号无效。
- DSTC 使 ERINT 中断信号无效。
- DSTC 设置所有 DQMSK[n] 寄存器，抑制 HW 传输请求。

尽管发出待机转换命令使 ERINT 中断信号无效，MONERS 寄存器值仍然保持不变。因此，如果待机转换命令强制停止传输，仍然可以通过读取 MONERS 寄存器检查传输信息。此外，只有 DSTC 处于正常状态时才能清除 MONERS 寄存器中的错误记录。待机释放命令让 DSTC 恢复正常状态后，通过 ERCLR 命令清除 MONERS 寄存器。

总线复位后，DQMSK[n] 寄存器中所有位的初始值为 0。如果发出待机转换命令，DQMSK[n] 中的所有位将设为 1。若要在 DSTC 恢复至正常状态后启动 HW 传输，在完成外设设置和 DES 重建后清除 HW 传输将使用的 DQMSK[n]。

Table 3-13 所示为各控制寄存器在各 DSTC 状态下的可访问性。 "O" 指该寄存器可以访问。"-" 指对该寄存器的访问被 DSTC 忽略且不起作用。"X" 指由于 DSTC 状态的改变，进程结果变为未定义。禁止执行标注 "X" 的访问。

在待机状态、转换状态 1 和转换状态 2 时，SWTR:SWDESP 写入值无法启动新的 SW 传输（对 SWTR 寄存器的写入被忽略）。

Table 3-13 各控制寄存器在不同 DSTC 状态下的可访问性

寄存器名称	寄存器访问	待机状态	正常状态	转换状态 1、2
CMD 寄存器	CMD 寄存器读取	O	O	O
	待机释放命令（写入）	O	-	-
	待机转换命令（写入）	-	O	-
	SWCLR / ERCLR / MKCLR 命令（写入）	-	O	X
HWDESP[n] 寄存器	读取	-	O	X
	写入	-	O	X
其他控制寄存器	读取	O	O	O
	写入	-	O	X

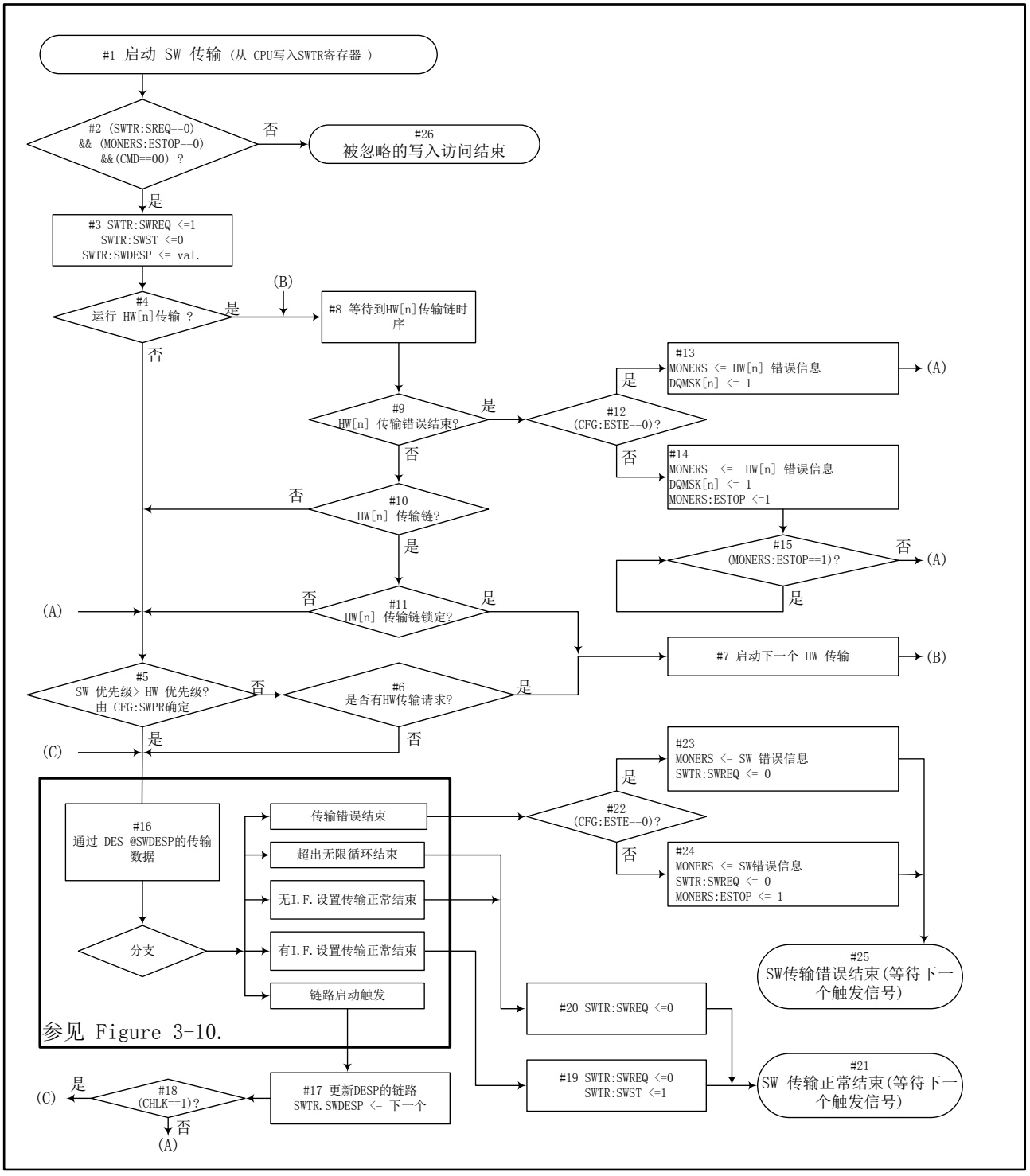
3.3 DSTC 操作流程

本节说明 DSTC 操作和操作流程图。

3.3.1 SW 传输流程

以下说明从 CPU 收到 SW 启动触发信号后 DSTC 执行的操作。Figure 3-8 所示为 DSTC 操作流程图。本图中的编号与图后说明中使用的编号对应。

Figure 3-8 DSTC 操作流程（SW 传输）



#1：通过 CPU 写入 SWTR 寄存器，开始 SW 启动传输。

#2：如果 SWTR 寄存器、MONERS 寄存器和 CMD 寄存器为 (SWTR:SWREQ==0)&&(MONERS:ESTOP==0)&&(CMD==00)，DSTC 将转到 #3。否则 DSTC 将转到 #26。

#3：将 SWTR:SWREQ 设为 "1" 并将 SWTR:SWST 清除为 0。将指定值储存在 SWTR:SWDESP 中。

#4 ~ #15 所示进程为判优器 2 的详细操作和其他 HW 传输进程。

#4：如果正在执行其他 HW 传输，DSTC 将转到 #8。否则 DSTC 将转到 #5。

#5：根据 CFG 寄存器中 SWPR 位的设置确定是 SW 传输还是 HW 传输更优先。如果 SW 传输更优先，DSTC 将转到 #16。否则 DSTC 将转到 #6。

#6：如果存在其他 HW 传输请求，DSTC 将转到 #7。否则 DSTC 将转到 #16。

#7：HW 传输的启动数据传输。

#8：在 HW 传输完成 DES 前，DSTC 保持等待启动执行 #1 发出的 SW 传输。

#9：如果该 HW 传输以错误的形式结束，DSTC 将转到 #12。否则 DSTC 将转到 #10。

#10：如果该 HW 传输中有链路启动传输，DSTC 将转到 #11。否则 DSTC 将转到 #5。

#11：如果该 HW 传输中的链路启动传输被锁定，DSTC 将转到 #7。否则 DSTC 将转到 #5。

#12：如果 CFG:ESTE 为 0，DSTC 将转到 #13。否则 DSTC 将转到 #14。

#13：如果 MONERS 寄存器中没有错误记录 (EST[2:0] = 000)，MONERS 寄存器将记录造成错误的 HW 传输 DES 的错误信息。如果存在错误记录 (EST[2:0] ≠ 000)，DSTC 将 DER 位设为 1。DSTC 将转到 #5。

#14：执行与 #13 相同的进程。同时，将 MONERS 寄存器中的 ESTOP 位设为 1。DSTC 将转到 #15。

#15：当 ESTOP 位为 1 时，DSTC 保持启动执行 #1 发出的 SW 传输。如果 CPU 向 CMD 寄存器发出 ERCLR 命令且 ESTOP 位被清除为 0，则 DSTC 将转到 #5。

#16：黑色粗线框内的流程说明根据 DESP 指定 DES 的 DSTC 传输操作。有关黑色粗线框内流程的详细说明，参见“指定 DESP 后的操作流程”一节。若为 SW 启动传输，DSTC 将根据 SWDESP 指定的 DES 执行传输。处理传输后，DSTC 操作分支为 Figure 3-10 所示的五项操作之一。

#17：若为链路启动，DSTC 将更新 SWTR:SWDESP 值。

#18：如果 CHLK 为 1，DSTC 将转到 #16 并相继执行链路启动的传输。否则 DSTC 将转到 #5。

#19：如果传输正常结束且有中断标志设置指令，DSTC 将执行 #19 所示处理。DSTC 将 SWTR:SWREQ 清除为 0 并将 SWTR:SWST 设为 1。

#20：如果传输正常结束且无中断标志设置指令，DSTC 将执行 #20 所示处理。DSTC 将 SWTR:SWREQ 清除为 0。

#21：DSTC 结束 #1 所示 SW 启动触发信号所启动的传输。DSTC 等待新启动触发信号或后续启动触发信号。已结束传输的 DES 的 DESP 保留在 SWTR:SWDESP 中。

#22：如果该 SW 启动传输因发生错误而结束且 CFG:ESTE 为 0，DSTC 将转到 #23。否则 DSTC 将转到 #24。

#23：如果 MONERS 寄存器中没有错误记录，MONERS 寄存器将记录造成错误的 SW 启动传输的 DES 错误信息。如果有错误记录，DSTC 将 DER 位设为 1。DSTC 将 SWTR:SWREQ 清除为 0。

#24：执行与 #23 相同的进程。同时，将 MONERS 寄存器中的 ESTOP 位设为 1，DSTC 保留其他 HW 传输的传输启动。

#25：#1 所示 SW 启动触发信号所导致的传输因发生错误而结束。不管 CHRS 值如何，SWTR:SWST 不被设为 1。 DSTC 等待新的启动触发信号。

#26：如 #1 所述，通过 CPU 写入 SWTR 寄存器时，如果未满足 #2 所示的条件，DSTC 将忽略对 SWTR 寄存器的写入。DSTC 不接受 SW 启动请求。

SW 传输时 DSTC 控制的其他信息

如果未满足 #2 所示的条件，则表示在 #2 之前已执行 SW 启动传输指令，且该传输未结束 (SWREQ ≠ 0) 或 DSTC 未处于正常状态 (CMD ≠ 00) 或 DSTC 处于错误停止状态 (ESTOP ≠ 0)，DSTC 将忽略且不接受 CPU 发出的 SW 启动请求。

注意 DSTC 的这种性态，特别是在 CFG:ESTE 设为 1 时使用 DSTC 的时候。如果 DSTC 已因另一个 HW 传输的错误而停止，任何新 SW 启动请求（写入寄存器）将被忽略，且 SWREQ 不设为 1。因此，如果写入 SWTR 寄存器后，DSTC 从 SWTR 寄存器的 SWREQ 位读取的值为 0，将无法确定是 SW 启动请求已被忽略还是传输已结束。此外，如果在 CFG:ESTE 设为 1 时使用 DSTC，SW 传输时，DES0:CHRS 设为 DSTC 未执行链路启动时的值，始终将 SWTR:SWST 设为 1。用这种方法设置 DES0:CHRS，写入 SWTR 寄存器时若 SWREQ 位和 SWST 位的读取值都为 0，表明未接受任何传输请求。如果未接受传输请求，由于 SWREQ 位或 SWST 位为 1，DSTC 可确定具体是 SW 启动请求已被忽略还是传输已结束。

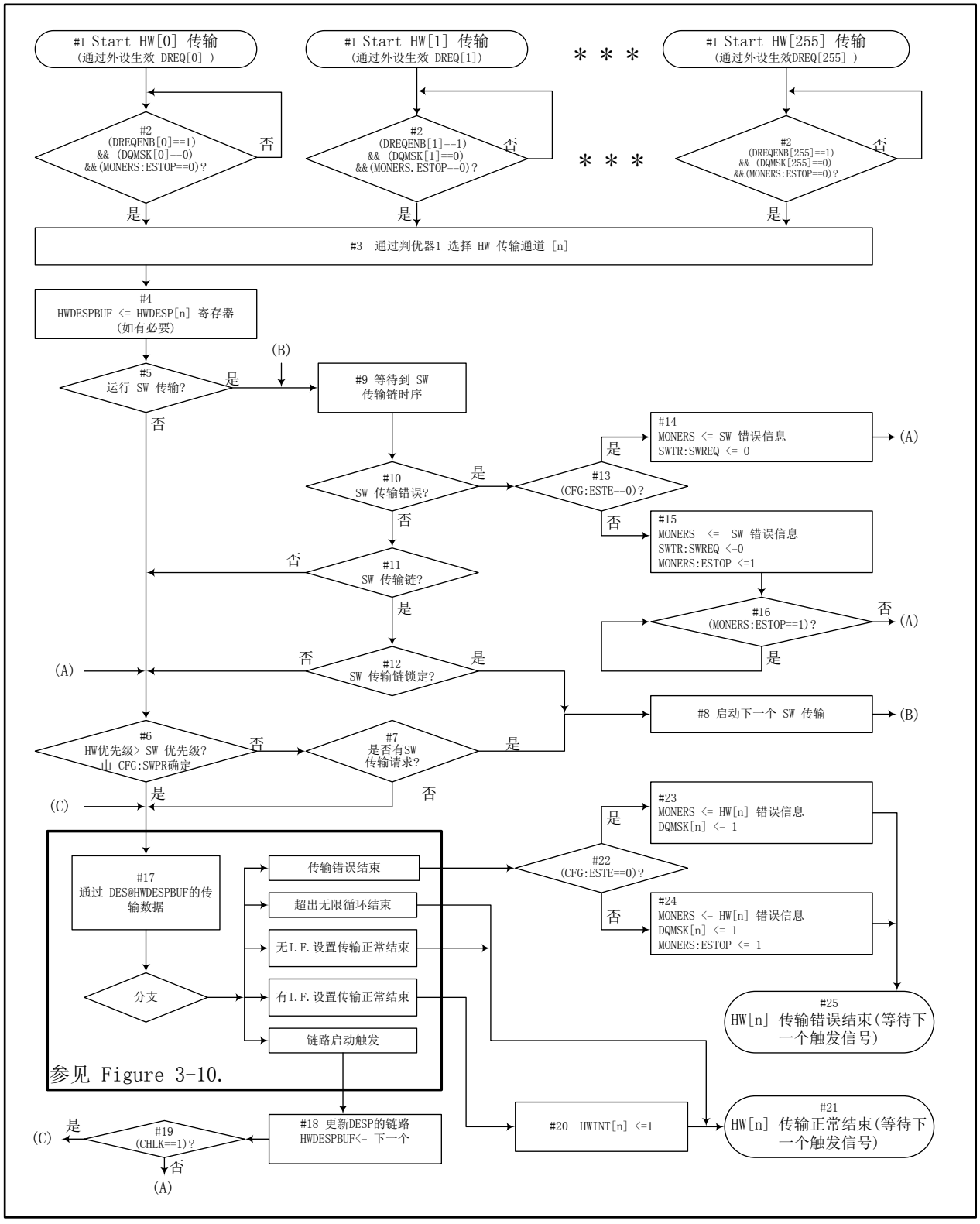
如果使用 SWST 位使 SWINT 中断信号生效，在中断处理进程中发出 SWCLR 命令将 SWST 清除为 0。即使 SWST 位被清除为 0，也可通过写入 SWTR 寄存器发出新的 SW 启动请求。然而，在 #3 所示的进程中，SWST 位始终被清除为 0 且 SWINT 中断信号始终被忽略。

在 #4 中，如果正在执行其他 HW 传输或已锁定该 HW 传输中的链路启动，则可能需要时间启动 SW 传输，即使 SW 传输具有更高优先级。

3.3.2 HW 传输流程

以下说明收到外设的 HW 启动触发信号后 DSTC 执行的操作。Figure 3-9 所示为 DSTC 操作流程图。本图中的编号与图后说明中使用的编号对应。

Figure 3-9 DSTC 操作流程（HW 传输）



#1：通过生效外设的 DREQ[n] 信号，DSTC 启动 HW 传输。

#2：如果 DREQENB[n] 寄存器、DQMSK[n] 寄存器和 MONERS 寄存器为 (DREQENB[n]==1)&& (DQMSK[n]==0)&&(MONERS:ESTOP==0)，则 DSTC 将转到 #3。如果 DQMSK[n] 寄存器或 ESTOP 寄存器设为 1，则 DSTC 将忽略外设的 DREQ[n] 信号并保持 HW 传输启动。

#3：#3 进程由判优器 1 执行。若有多条 HW 通道的传输请求，DSTC 将选择执行传输的 HW 通道编号 (n)。DSTC 保持其他通道发出的传输请求，直到所选 HW[n] 传输正常结束或因错误结束为止，然后 DSTC 启动等待启动触发信号。

#4：基于所选通道编号 (n)，将 HWDESP[n] 寄存器的 DESP 值储存在 HWDESPBUF 中。如果通道编号与前文中的某项编号相同，HWDESPBUF 值有效，则跳转访问 HWDESP[n] 寄存器。

#5 ~ #16 所示进程为有关判优器 2 操作和其他 SW 传输进程的参数。DSTC 执行与 SW 传输流程 #4 ~ #15 所示相同的进程。如果存在 SW 启动请求且该 SW 传输具有更高优先级，或如果 SW 传输中的链路启动已被锁定，则 DSTC 将首先执行 SW 传输。此外，如果 SW 传输因错误而结束，DSTC 将把 SW 传输的错误信息记录在 MONERS 寄存器中。如果因 SW 传输的错误停止而将 ESTOP 位设为 1，则 DSTC 保持 HW[n] 传输启动。

#17：黑色粗线框内的流程说明根据 DESP 指定 DES 进行的 DSTC 传输操作。有关黑色粗线框内流程的详细说明，参见 “指定 DESP 后的操作流程” 一节。若为 HW 传输，DSTC 将根据 HWDESP 指定的 DES 执行传输。处理传输后，DSTC 操作分支成 Figure 3-10 所示的五项操作之一。

#18：若为链路启动，DSTC 将更新 HWDESPBUF。

#19：如果 CHLK 为 1，DSTC 转到 #17 并相继执行链路启动的传输。否则 DSTC 将转到 #6。

#20：如果传输正常结束且有中断标志设置指令，DSTC 将 HWINT[n] 设为 1。

#21：DSTC 结束 #1 所示 HW 启动触发信号所引起的传输。DSTC 等待新启动触发信号或后续启动触发信号。如果 DSTC 保持 #3 中的其他通道的 HW 启动触发信号，则判优器 1 将选择将执行传输的通道且 DSTC 转到 #4。

#22：如果该 HW 启动传输因发生错误而结束且 CFG:ESTE 为 0，DSTC 转到 #23。否则 DSTC 转到 #24。

#23：如果 MONERS 寄存器中没有错误记录，MONERS 寄存器将记录正执行 HW 传输的 DES 错误信息。如果有错误记录，DSTC 将 DER 位设为 1。此外，DSTC 将 DQMSK[n] 寄存器设为 1，抑制 HW 通道 n 发出后续传输请求。

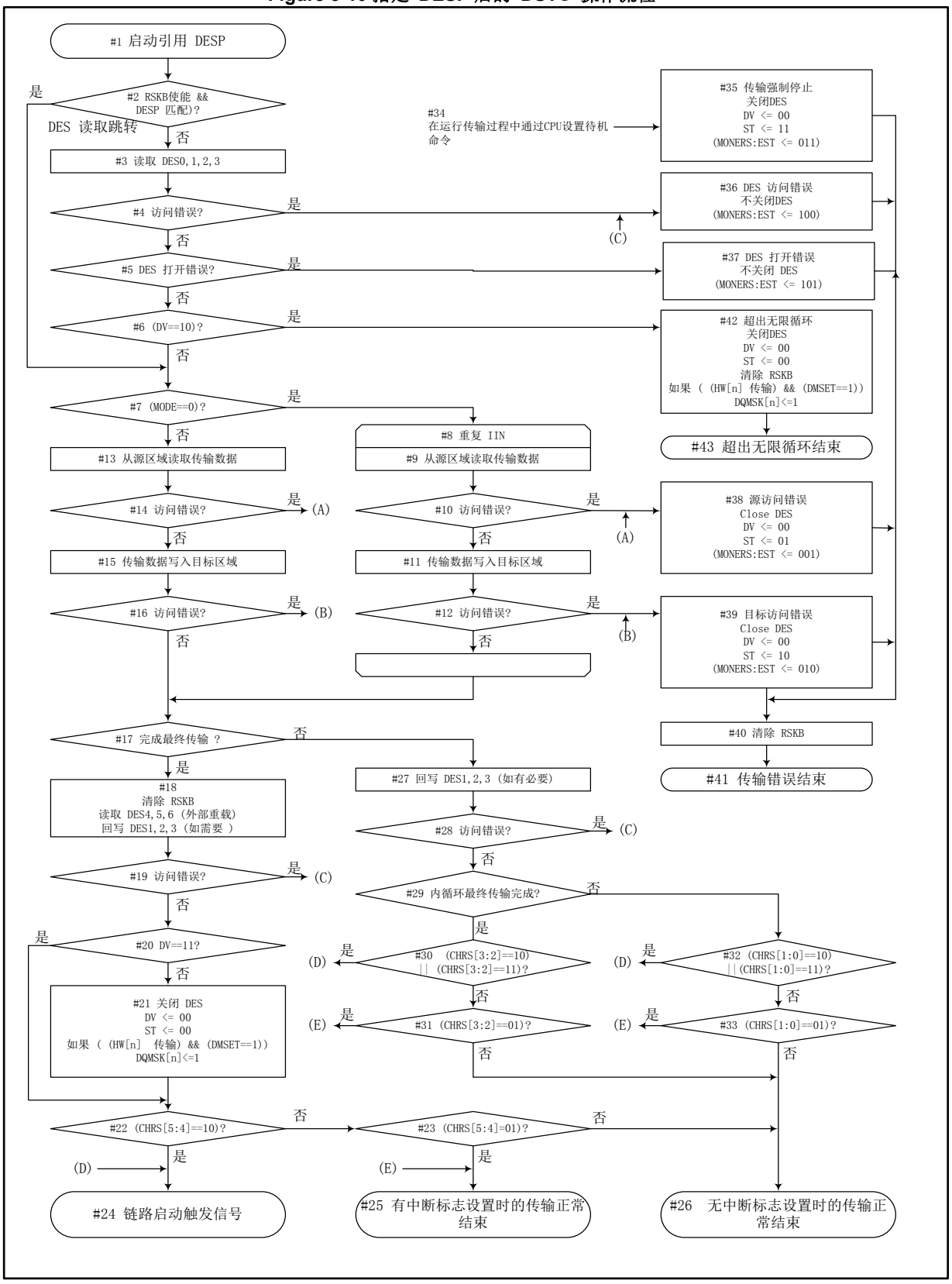
#24：执行与 #23 相同的进程。同时，将 MONERS 寄存器中的 ESTOP 位设为 1，DSTC 保持其他启动传输的启动。

#25： #1 所示 HW 启动触发信号所引起的传输因发生错误而结束。不管 CHRS 值如何，HWINT[n] 寄存器不设为 1。 DSTC 等待新的启动触发信号。如果 DSTC 保持其他通道的 HW 启动触发信号，则判优器 1 选择将执行传输的通道且 DSTC 转到 #4。

3.3.3 指定 DESP 后的操作流程

以下说明已执行 DESP 后 DSTC 执行的操作。Figure 3-10 所示为 DSTC 在指定 DESP 后执行的操作流程图。本图中的编号与图后说明中使用的编号对应。

Figure 3-10 指定 DESP 后的 DSTC 操作流程



#1：DSTC 启动操作，访问判优器 2 指定的 DESP。

#2：如果使能读取跳转缓冲功能 (CFG:RBDIS = 0) 且 DSTC 访问的 DESP 与其之前访问的 DESP 相同，则 DSTC 跳过对内存区域 DES 的访问并转到 #7。否则 DSTC 转到 #3。

#3, #4：DSTC 读取指定 DESTP+DESP 区域内的 DES。如果访问 DES 区域时发生访问错误，DSTC 转到 #36。否则 DSTC 转到 #5。

#5：DSTC 检验 DES 的参数。如果发生 DES 打开错误，DSTC 转到#37。否则 DSTC 转到 #6。有关 DES 打开错误的详细说明，参见 "3.2.8MONERS 寄存器"。

#6：如果 DES0:DV 为 10，DSTC 转到 #42。否则 DSTC 转到 #7。

#7：如果 DES0:MODE 为 0，DSTC 转到 #8。否则 DSTC 转到 #13。

#8 ~ #12：如果指定模式 0 传输，DSTC 按 DES1:IIN 指定的次数相继执行传输。如果在传输源访问时发生访问错误，DSTC 转到 #38。如果在传输目标访问时发生出现访问错误，DSTC 转到 #39。如果在传输源访问或传输目标访问时没有发生任何访问错误，DSTC 转到 #17。

#13 至 #16：如果指定模式 1 传输，DSTC 执行一次传输。如果在传输源访问时发生访问错误，DSTC 转到 #38。如果在传输目标访问时发生访问错误，DSTC 转到 #39。如果在传输源访问或传输目标访问时没有发生任何访问错误，DSTC 转到 #17。

#17：如果 IIN×ORM 次传输已结束 (ORM ==1 且 IRM ==1)，DSTC 转到 #18。否则 DSTC 转到 #27。

#18, #19：DSTC 清除读取跳转缓冲器，根据 DES0:ORL 中的指令从 DES4 ~ DES6 区读取要求值并将要求值写入 DES1 ~ DES3 区域。如果更新 DES 区域引用时发生访问错误，DSTC 转到 #36。否则 DSTC 转到 #20。

#20：如果 DES0:DV 为 11，DSTC 转到 #22。如果 DES0:DV 为 01，DSTC 转到 #21。（在 #5 中，如果 DES0:DV 为 00，DSTC 转到 #37。在 #6 中，如果 DES0.DV 为 10，DSTC 转到 #42）。

#21：DSTC 执行 DES 关闭进程。 DSTC 更新 DES0:DV 为 00 并更新 DES0:ST 为 00。在 HW 启动触发信号引起的传输（或 HW 启动触发信号引起的链路传输）中，DES0:DMSET 为 1 时，DSTC 将 DQMSK[n] 设为 1 并抑制通道 n 发出的后续 HW 启动传输请求。

#22, #23：如果 DES0:CHRS[5:4] 为 10，DSTC 转到 #24。如果 DES0:CHRS[5:4] 为 01，DSTC 转到 #25。如果 DES0:CHRS[5:4] 为 00，DSTC 转到 #26。（在 #5 中，如果 DES0:CHRS[5:4] 为 11，DSTC 转到 #37）。

#24：DSTC 执行链路启动传输。 #1 所示 DESP 的传输正常结束。有关后续将执行的操作，参见前述章节。

#25：#1 所示 DESP 的在设置有中断标志时传输正常结束。有关后续将执行的操作，参见前述章节。

#26：#1 所示 DESP 的在没有设置中断标志时传输正常结束。有关后续将执行的操作，参见前述章节。

#27, #28：DSTC 将下一项启动触发信号引起的传输所需的值反写至 DES1 ~ DES3 区域。如果更新 DES 区域时发生访问错误，DSTC 转到 #36。否则 DSTC 转到 #29。

#29：如果 MODE 为 0，DSTC 转到 #30。如果 MODE 为 1 且已按照内循环计数次数结束传输 (ORM !=1 且 IRM ==1)，DSTC 转到 #30。否则 DSTC 转到 #32。

#30, #31：如果 DES0:CHRS[3:2] 为 10 或 11，DSTC 转到 #24。如果 DES0:CHRS[3:2] 为 01，DSTC 转到 #25。如果 DES0:CHRS[3:2] 为 00，DSTC 转到 #26。

#32, #33：如果 DES0:CHRS[1:0] 为 10 或 11，DSTC 转到 #24。如果 DES0:CHRS[1:0] 为 01，DSTC 转到 #25。如果 DES0:CHRS[1:0] 为 00，DSTC 转到 #26。

#34：如果 CPU 传输时发出待机转换命令，DSTC 中断传输并转到 #35。

#35：如果传输被强制停止，DSTC 执行 DES 错误关闭进程。DSTC 更新 DES0:DV 为 00 并更新 DES0:ST 为 11，然后转到 #40。如果 MONERS 寄存器中没有任何错误记录，DSTC 设 MONERS:EST 为 011。

#36：如果发生 DES 访问错误，DSTC 不执行 DES 错误关闭进程，并转到 #40。如果 MONERS 寄存器中没有任何错误记录，DSTC 设 MONERS:EST 为 100。

#37：如果发生 DES 打开错误，DSTC 不执行 DES 错误关闭进程，并转到 #40。如果 MONERS 寄存器中没有任何错误记录，DSTC 设 MONERS:EST 为 101。

#38：如果发生传输源访问错误，DSTC 执行 DES 错误关闭进程。 DSTC 更新 DES0:DV 为 00 并更新 DES0:ST 为 01，然后转到 #40。 如果 MONERS 寄存器中没有任何错误记录，DSTC 设 MONERS:EST 为 001。

#39：如果发生传输目标访问错误，DSTC 执行 DES 错误关闭进程。DSTC 更新 DES0:DV 为 00 并更新 DES0:ST 为 10，然后转到 #40。 如果 MONERS 寄存器中没有任何错误记录，DSTC 设 MONERS:EST 为 010。

#40, #41：DSTC 清除读取跳转缓冲器。 #1 所示 DESP 传输因发生错误而结束。有关后续将执行的操作，参见前述章节。

#42, #43：DSTC 清除读取跳转缓冲器。 DSTC 执行 DES 关闭进程。 DSTC 更新 DES0:DV 为 00 并更新 DES0:ST 为 00。在 HW 启动触发信号引起的传输（或 HW 启动触发链路传输）中，DES0:DMSET 为 1 时，DSTC 设 DQMSK[n] 为 1 并抑制后续 HW 启动传输请求。#1 所示 DESP 的传输因退出无限循环而结束。有关后续将执行的操作，参见前述章节。

4. DSTC 操作和控制示例

本节说明 DSTC 操作和控制示例。

4.1. 传输操作示例 1
4.2. 传输操作示例 2
4.3. 传输操作示例 3
4.4. 传输操作示例 4
4.5. 传输操作示例 5
4.6. DSTC 控制示例

4.1 传输操作示例 1

本节说明传输操作示例 1。传输操作示例 1 为模式 0 中的 SW 传输示例。

传输启动时的 DES 值

Table 4-1 所示为传输操作示例 1 中的 DES 设置。设置 ORL[2:0] 为 101 时，没有 DES5 区域。 DES 为 6 字配置，包含 DES0 ~ DES4 和 DES6。（DES6 地址为 DESP+0x0014。）

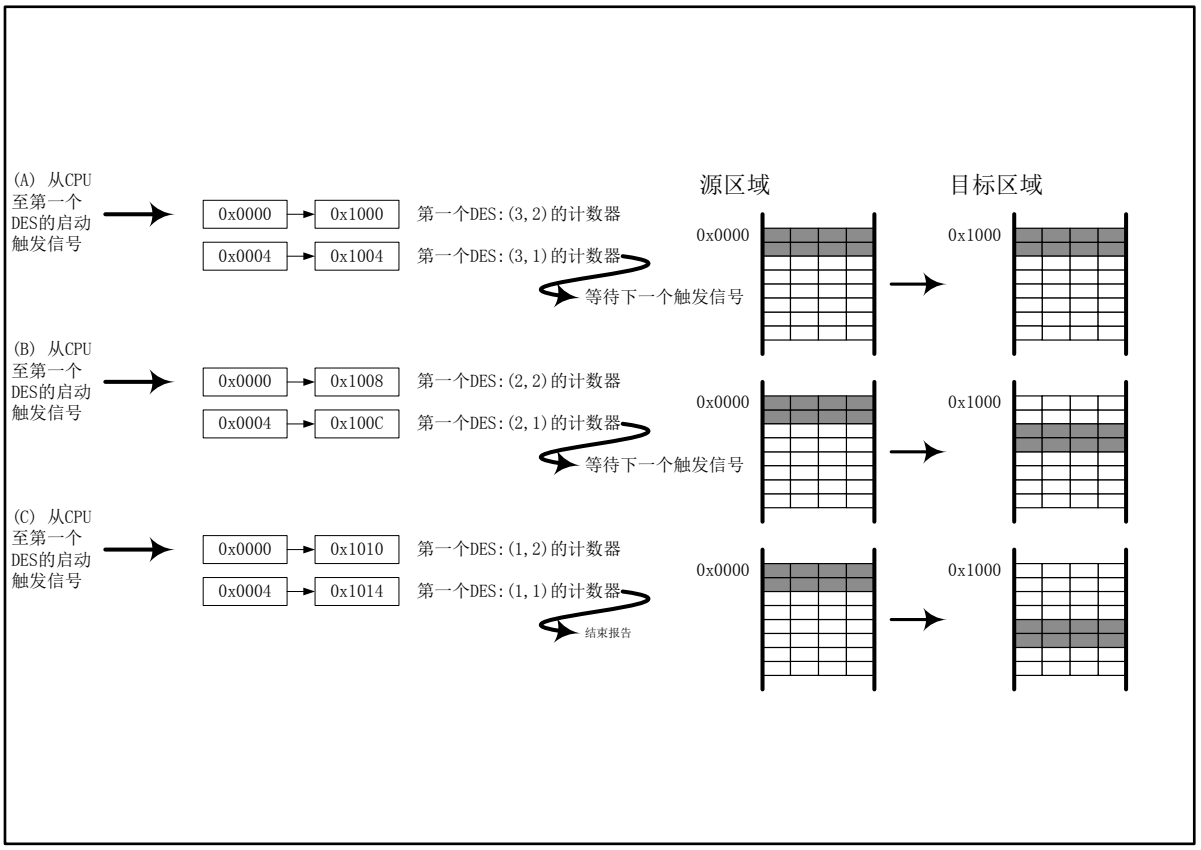
Table 4-1 传输操作示例 1 中传输启动时的 DES 值

地址	DES 编号	值
DESP+0x0000	DES0	DES0 =0x901406A1 DV = 01：在传输结束时执行 DES 关闭进程 MODE = 0，TW = 10：模式 0，32 位（字）传输 ORL = 101：外部重载： DES1 <=DES4，DES3 <= DES6 SAC = 001：递增 TW×1，需要内部重载 DAC = 000：递增 TW×1，无需内部重载 CHRS = 010100：无链路启动，已设置中断标志。 DMSET = 0：传输为 SW 启动传输时，设 DMSET 为 0。 CHLK = 0：无链路锁定 ACK = 00：传输为 SW 启动传输时，设 ACK 为 00。 PCHK=1001：奇偶校验
DESP+0x0004	DES1	ORM = 0x0002，IIN = 0x0003
DESP+0x0008	DES2	SA = 0x00000000
DESP+0x000C	DES3	DA = 0x00001000
DESP+0x0010	DES4	ORM = 0x0002，IIN = 0x0003（与 DES1 相同）
DESP+0x0014	DES6	DA = 0x0000 1000（与 DES3 相同）

传输操作流程

Figure 4-1 所示为传输操作示例 1 的传输操作流程。图中启动触发信号 (A) 和 (B) 所示为通过 CPU 将 DESP 写入 SWTR 寄存器。

Figure 4-1 传输操作示例 1 操作流程



DSTC 因启动触发信号 (A) 启动 DES 传输。如 Figure 4-1 所示，矩形框内的值指传输原地址和传输目标地址。DSTC 启动 32 位传输至 0x0000 ~ 0x1000 地址区域。DSTC 相继执行三次 (IIN = 3) 32 位传输。关于 DES 传输数量计数器，启动传输时，外循环计数器剩余值 (ORM) 为 2，内循环计数器剩余值 (IRM) 值为 3。如图 Figure 4-1 所示，传输数量计数器剩余值表示为 (2,3)。三次传输后，传输数量计数器读取 (2,1)。ORM 不为 1 而 IRM 为 1 时，IRM 使用 CHRS[3:2] 值确定下一进程。CHRS[3:2] 为 01 时，DSTC 设 SWTR:SWST 为 1 并等待下一个启动触发信号。

DSTC 启动由启动触发信号 (B) 再次触发的 DES 传输。地址的内部重载适用于 SA。DA 保持增量。DSTC 重新启动 32 位传输至 0x0000 ~ 0x100C 地址区域。DSTC 相继执行三次 (IIN = 3) 传输。三次传输后，传输数量计数器从 (1,3) 开始计数并读取 (1,1)。 ORM 为 1 且 IRM 为 "1" 时，DSTC 执行 DES 关闭进程，同时设 DES 的 DV 为 01。DSTC 使用 CHRS[5:4] 值确定下一进程。CHRS[5:4] 为 01 时，DSTC 设 SWTR:SWST 为 1。

传输结束后储存的 DES 值

如果传输操作示例 1 的传输正常结束，如 Table 4-2 所示更新 DES 值。表中黑体字表示与传输启动前不相同的值。根据 ORL 设置，将 DES4 和 DES6 的值分别复制到 DES1 和 DES3，使 DES1 和 DES3 与启动传输前的值相同。尽管 OR[1] 为 0，根据内部重载设置，DES2 的值仍与启动传输前相同。DSTC 更新 DV 值为 00，并将 DES 所有权归还 CPU。DSTC 更新 ST 值为 00，并通知 CPU 传输已正常结束。

使用上述更新的 DES 执行传输，需通过 CPU 更新 DV 值。

Table 4-2 传输操作示例 1 中传输结束后的 DES 值

DES 编号	值
DES0	**DV = 00，ST = 00，**其他值与启动传输时的值相同
DES1	与启动传输时相同。
DES2	与启动传输时相同。
DES3	与启动传输时相同。
DES4	与启动传输时相同。
DES6	与启动传输时相同。

DSTC 执行的 DES 操作

在传输操作示例 1 中，通过 DSTC 进行的 DES 操作如下：

在启动触发信号 (A) 之后：

DSTC 从 DES0 读取指令。
DSTC 从 DES1 读取 (2,3)。
DSTC 从 DES2 读取 0x0000。
DSTC 从 DES3 读取 0x1000。

第一次传输后：

DSTC 反写 (1,3) 至 DES1。
DSTC 不反写 DES2，所以相同值。
DSTC 反写 0x100C 至 DES3。

在启动触发信号 (B) 之后：

* DSTC 从 DES0 读取指令。
DSTC 从 DES1 读取 (1,3)。
* DSTC 从 DES2 读取 0x0000。
* DSTC 从 DES3 读取 0x100C。

第二次传输后：

DSTC 从 DES4 复制 (2,3) 至 DES1，以进行外部重载。
DSTC 不反写 DES2，所以相同值。
DSTC 从 DES6 复制 0x1000 至 DES3，以进行外部重载。
DSTC 反写 DES0，关闭 DES。

如果在启动触发信号 (A) 和启动触发信号 (B) 之间没有发出其他传输请求，第二次传输时，DSTC 将使用读取跳转缓冲器中的 DES0,1,2,3 的值。因此，上述具有 * 标记的操作将被跳过。有关详细说明，参见 3.3.3 指定 DESP 后的操作流程。

补充信息

在启动触发信号 (A) 所触发的传输结束且 DSTC 进入启动触发等待状态前，不会发出启动触发信号 (B)。SWTR:SWREQ 为 1 时，写入（SW 启动触发信号） SWTR 寄存器将被忽略。

如果启动触发信号 (A) 所触发的传输结束且 DSTC 进入启动触发等待状态，可发出启动触发信号 (B) 之外的其他 DES 的 SW 启动触发信号。其他 DES 的 SW 传输结束后，如果该 DES 的 DESP 被写入 SWTR 寄存器，将发出启动触发信号 (B)，且 DSTC 将继续从传输 (A) 结束之处开始数据传输。

启动触发信号 (A) 所触发的传输结束后，并不总是需要启动触发信号 (B)。如果未发出启动触发信号 (B) 且 DSTC 不继续传输数据，启动触发信号 (A) 所触发的传输结束后（即使未执行 DES 关闭进程），CPU 可修改传输的 DES 区域。

4.2 传输操作示例 2

本节说明传输操作示例 2。传输操作示例 2 为模式 1 的 HW 传输示例。

传输启动时的 DES 值

Table 4-3 所示为传输操作示例 2 的 DES 设置。ORL[2:0] 设为 000 时，没有 DES4 区域、DES5 区域或 DES6 区域。 DES 为 4 字配置，包含 DES0 ~ DES3。

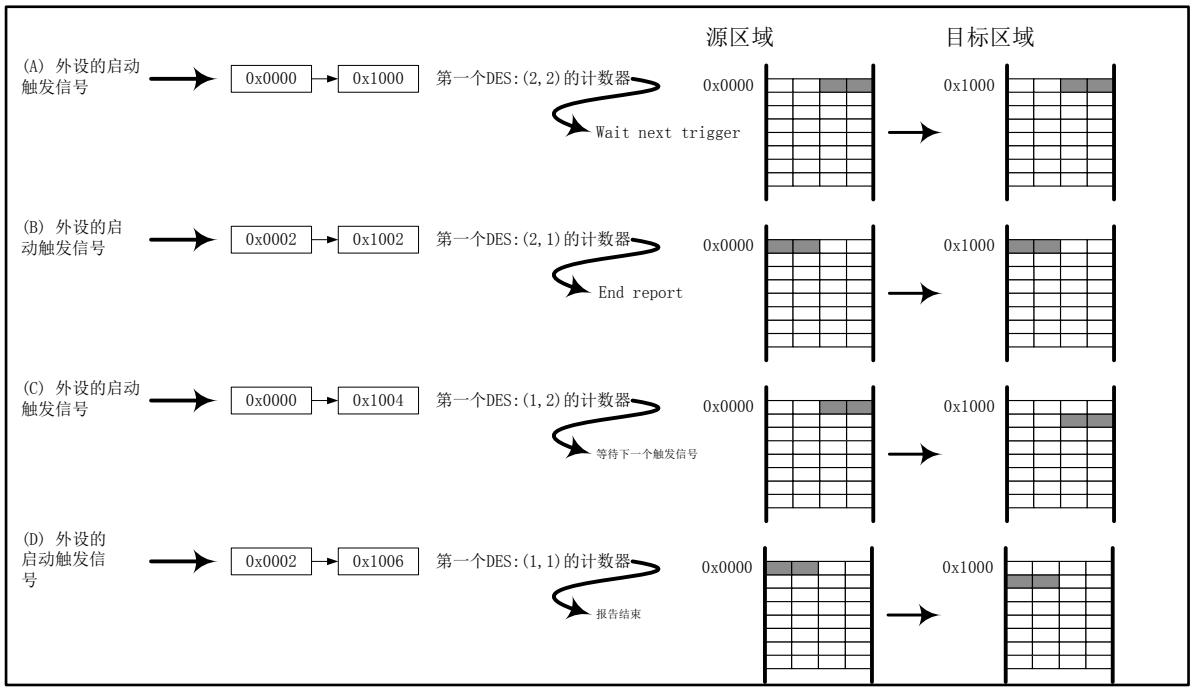
Table 4-3 传输操作示例 2 中传输启动时的 DES 值

地址	DES 编号	值
DESP+0x0000	DES0	DES0 = 0x01140511 DV = 01：传输时的 DES 关闭进程 MODE = 1，TW = 01：模式 1，16 位（半字）传输 ORL = 000：无需外部重载 SAC = 001：递增 TW×1，需要内部重载 DAC = 000：递增 TW×1，无需内部重载 CHRS = 010100：无链路启动，已设置中断标志。 DMSET = 0：在 DES 关闭进程中不设置 DQMSK[n] CHLK = 0：无链路锁定 ACK = 01：通过 DREQ 信号直接启动 DES 时，将 ACK 设为 01。 PCHK = 0000：奇偶校验
DESP+0x0004	DES1	ORM = 0x0002, IIN = 0x02, IRM = 0x02
DESP+0x0008	DES2	SA = 0x00000000
DESP+0x000C	DES3	DA = 0x00001000

传输操作流程

Figure 4-2 所示为传输操作示例 2 的传输操作流程。如 Figure 4-2 所示，启动触发信号 (A) 与生效的外设 DREQ[n] 信号相对应。对于 DREQ[n]，启动传输前设置 DREQENB[n]、DQMSK[n] 和 HWDESP[n] 寄存器值。

Figure 4-2 传输操作示例 2 操作流程



DSTC 启动由启动触发信号 (A) 所触发的 DES 的传输。DSTC 执行一次 16 位传输至 0x0000 ~ 0x1000 地址区域。传输数量计数器读取 (2,2)。 IRM 不为 1 时，根据 DES 的 CHRS[1:0] 设置 (CHRS[1:0] = 00)，DSTC 不设置 HWINT[n] 寄存器为 1。DSTC 等待下一个启动触发信号。

DSTC 再次启动由启动触发信号 (B) 所触发的 DES 传输。DSTC 执行一次 16 位传输至 0x0002 ~ 0x1002 地址区域。传输数量计数器读取 (2,1)。 ORM 不为 1 而 IRM 为 1 时，根据 DES 的 CHRS[3:2] 设置 (CHRS[3:2] = 01)，DSTC 设置 HWINT[n] 寄存器为 1。DSTC 等待下一个启动触发信号。

DSTC 再次启动由启动触发信号 (C) 所触发的 DES 传输。地址的内部重载适用于 SA。DA 保持增量。DSTC 执行一次 16 位传输至 0x0000 ~ 0x1004 地址区域。传输数量计数器读取 (1,2)。IRM 不为 "1" 时，根据 DES 的 CHRS[1:0] 设置 (CHRS[1:0] = 00)，DSTC 不设置 HWINT[n] 寄存器为 "1"。DSTC 等待下一个启动触发信号。

DSTC 再次启动由启动触发信号 (D) 所触发的 DES 传输。DSTC 执行一次 16 位传输至 0x0002 ~ 0x1006 地址区域。传输数量计数器读取 (1,1)。 ORM 为 1 且 IRM 为 "1" 时，DSTC 执行 DES 关闭进程，同时将 DES 的 DV 设为 01。根据 DES 的 CHRS[5:4] 设置 (CHRS[5:4] = 01)，DSTC 设 HWINT[n] 寄存器为 1。

传输结束后储存的 DES 值

传输操作示例 2 的传输正常结束时，如 Table 4-4 所示更新 DES 值。表中黑体字表示与传输启动前不相同的值。已将 ORL 设为 "000" 时，如果表中的值用于下一个传输，DSTC 不能执行与上述传输相同的传输。在这种情况下，需通过 CPU 重建 DES。

Table 4-4 传输操作示例 2 中传输结束后的 DES 值

DES 编号	值
DES0	DV = 00，ST = 00，其他值与启动传输时的值相同。
DES1	ORM = 0x0001，IRM = 0x01，IIN 与启动传输时相同。
DES2	与启动传输时相同。
DES3	DA = 0x00001006

4.3 传输操作示例 3

本节说明使用后续 DES 的链路启动相关传输操作示例 3。

传输启动时的 DES 值

在传输操作示例 3 中，DSTC 使用链路启动重组 0x0000 至 0x00FF 的数据并将数据传输至 0x0100 和 0x01FF 之间的区域。在本示例中使用四个 DES。第一个 DES 称为 1st-DES，第二个 DES 称为 2nd-DES，第三个 DES 称为 3rd-DES，第四个 DES 称为 4th-DES。Table 4-5 详细说明四个 DES。在各 DES 中均没有 DES4。Table 4-5 没有列出四个 DES 的地址。但是在内存上按照从第一个 DES 到第四个 DES 的顺序依次排列四个 DES。

Table 4-5 传输操作示例 3 中传输启动时的 DES 值

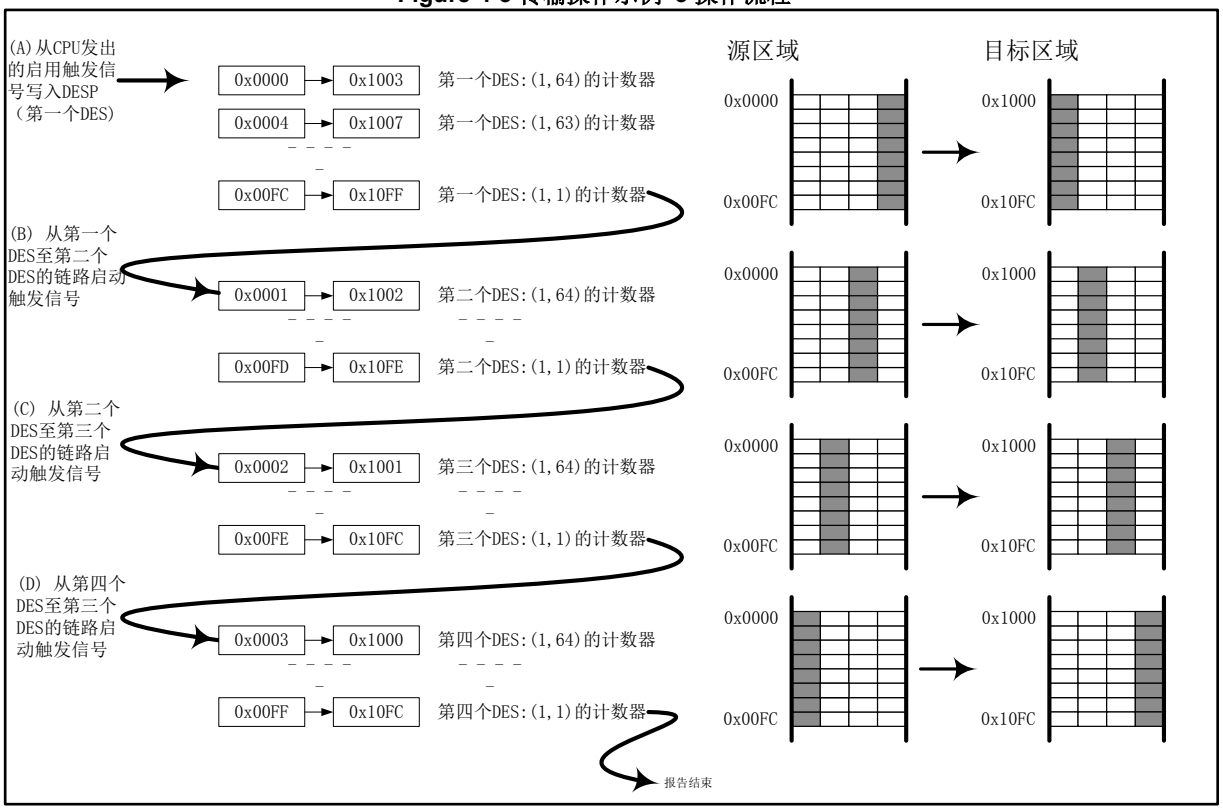
区	DES 编号	值
1st-DES	DES0	DES0 = 0x702090C3 DV = 11：在传输结束时不执行 DES 关闭进程 MODE = 0，TW = 00：模式 0，8 位（字节）传输 ORL = 110：外部重载 DES2<=DES5，DES3<= DES6 SAC = 100：递增 TW×4，无需内部重载 DAC = 100：递增 TW×4，无需内部重载 CHRS = 100000：在后续 DES 中有链路启动；未设置中断标志。 DMSET = 0：传输为 SW 传输时，设 DMSET 为 "0"。 CHLK = 0：无链路锁定 ACK = 00：传输为 SW 传输时，设 ACK 为 "00"。 PCHK = 0111：奇偶校验
	DES1	ORM = 0x0001，IIN = 0x0040
	DES2	SA = 0x00000000
	DES3	DA = 0x00001003
	DES5, DES6	DES5 的值与第一个 DES 中 DES2 的值相同，DES6 的值与第一个 DES 中 DES3 的值相同。
2nd-DES	DES0	与第一个 DES 中 DES0 相同
	DES1	与第一个 DES 中 DES1 相同
	DES2	SA = 0x00000001
	DES3	DA = 0x00001002
	DES5, DES6	DES5 的值与第二个 DES 中 DES2 的值相同，DES6 的值与第二个 DES 中 DES3 的值相同。
3rd-DES	DES0	与第一个 DES 中 DES0 相同
	DES1	与第一个 DES 中 DES1 相同
	DES2	SA = 0x00000002
	DES3	DA = 0x00001001
	DES5, DES6	DES5 的值与第三个 DES 中 DES2 的值相同，DES6 的值与第三个 DES 中 DES3 的值相同。

区	DES 编号	值
4th-DES	DES0	DES0 = 0x401090C3 CHRS = 010000：无链路启动，已设置中断标志。 PCHK = 0100：奇偶校验 其他值与第一个 DES 中 DES0 的值相同
	DES1	与第一个 DES 中 DES1 相同
	DES2	SA = 0x00000003
	DES3	DA = 0x00001000
	DES5, DES6	DES5 的值与第四个 DES 中 DES2 的值相同，DES6 的值与第四个 DES 中 DES3 的值相同。

传输操作流程

Figure 4-3 所示为传输操作示例 3 的传输操作流程。Figure 4-3 中启动触发信号 (A) 所示为通过 CPU 将第一个 DES 的 DESP 写入 SWTR 寄存器。Figure 4-3 中的启动触发信号 (B)，(C) 和 (D) 为链路启动触发。

Figure 4-3 传输操作示例 3 操作流程



DSTC 启动由启动触发信号 (A) 所触发的第一个 DES 传输。DSTC 执行一次 8 位传输至 0x0000 ~ 0x1003 地址区域，以及另一次 8 位传输至 0x0004 ~ 0x1007 地址区域。DSTC 按照上述顺序相继执行 64 次 (IIN = 64) 传输。64 次传输后，第一个 DES 的传输数量计数器从 (1,64) 开始计数并读取 (1,1)。当第一个 DES 的 DV 为 11 时，DSTC 不执行第一个 DES 相关的 DES 关闭进程。第一个 DES 的 CHRS[5:4] 为 10 时，将发出后续第二个 DES 传输相关的链路启动触发信号。

DSTC 启动由链路启动触发信号 (B) 所触发的第二个 DES 的传输。执行 8 位传输 64 次后，计数器值变成 (1,1)。第二个 DES 的 DV 为 11 时，DSTC 不执行第二个 DES 相关的 DES 关闭进程。第二个 DES 的 CHRS[5:4] 为 10 时，发出后续第三个 DES 传输相关的链路启动触发信号。

DSTC 启动由启动触发信号 (C) 所触发的第三个 DES 传输。执行 8 位传输 64 次后，计数器值变成 (1,1)。第三个 DES 的 DV 为 11 时，DSTC 不执行第三个 DES 相关的 DES 关闭进程。第三个 DES 的 CHRS[5:4] 为 10 时，发出后续第四个 DES 传输相关的链路启动触发信号。

DSTC 启动由启动触发信号 (D) 所触发的第四个 DES 传输。执行 8 位传输 64 次后，计数器值变成 (1,1)。第四个 DES 的 DV 为 11 时，DSTC 不执行第四个 DES 相关的 DES 关闭进程。第四个 DES 的 CHRS[5:4] 为 01 时，DSTC 设 SWTR:SWST 为 1 并结束传输。

如传输操作示例 3 所述，如果使用链路启动功能，只需向第一个 DES 发出启动触发信号，便可逐个自动执行多单元 DES 所定义的传输。

传输结束后储存的 DES 值

传输操作示例 3 中的传输正常结束时，根据 ORL 和 DV 的设置（ORL = 0 或 1，DV = 11），所有 DES 区域中的 DES 值与启动传输前的值相同。由于 DSTC 未执行 DES 关闭进程，所以不更新 DV 或 ST。在下一个传输中，只发出启动触发信号便可执行与先前传输相同的传输。

4.4 传输操作示例 4

本节说明传输操作示例 4。

传输启动时的 DES 值

传输操作示例 4 的详细说明如下。本示例说明 CHRS 设为 11 时，再次执行当前 DES 的链路启动。Table 4-6 所示为 DES 设置。

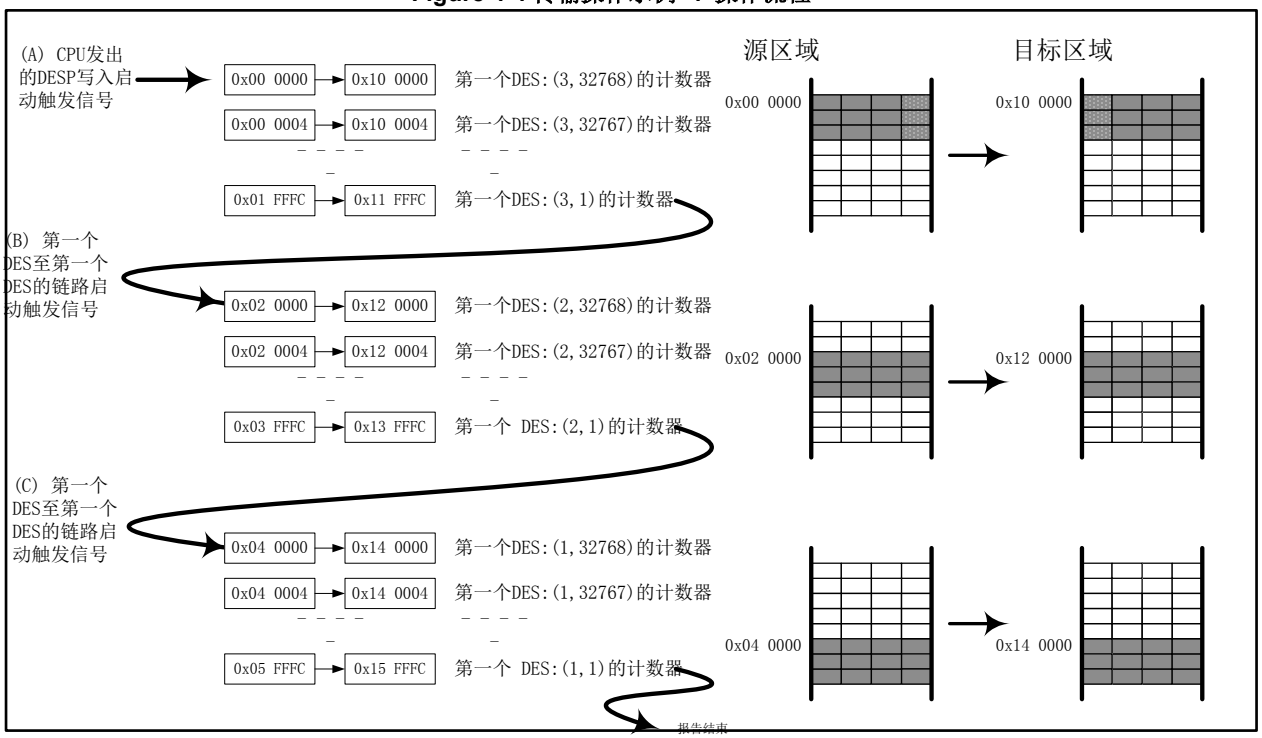
Table 4-6 传输操作示例 4 中传输启动时的 DES 值

DES 编号	值
DES0	DES0 = 0xF01C0201 DV = 01：将在传输结束时执行 DES 关闭进程 MODE = 0，TW = 10：模式 0，32 位（字）传输 ORL = 000：无需外部重载 SAC = 000：递增 TW×1，无需内部重载时 DAC = 000：递增 TW×1，无需内部重载 CHRS = 011100：在当前 DES 中有链路启动；已设置中断标志。 DMSET = 0：传输为 SW 启动传输时，设 DMSET 为 "0"。 CHLK = 0：无链路锁定 ACK = 00：传输为 SW 启动传输时，设 ACK 为 "00"。 PCHK = 1111：奇偶校验
DES1	ORM = 0x0003，IIN = 0x8000
DES2	SA = 0x00000000
DES3	DA = 0x00100000

传输操作流程

Figure 4-4 所示为传输操作示例 4 中的传输操作流程。Figure 4-4 中启动触发信号 (A) 表示通过 CPU 将 DESP 写入 SWTR 寄存器。Figure 4-4 中的启动触发信号 (B) 和 (C) 为链路启动触发。

Figure 4-4 传输操作示例 4 操作流程



DSTC 启动由启动触发信号 (A) 所触发的 DES 传输。DSTC 相继执行 32 位传输 32768 次 (IIN = 32768)，传输时递增地址。32768 次传输后 DES 传输数量计数器从 (3,32768) 开始计数并读取 (3,1)。DES 的 CHRS[3:2] 为 11 时，将再次发出相同 DES 传输的链路启动触发信号。

DSTC 启动由链路重启触发信号 (B) 再次触发的 DES 传输。DSTC 再次执行 32768 次传输。32768 次传输后 DES 传输数量计数器读取 (2,1)。DES 的 CHRS[3:2] 为 11 时，将再次发出相同 DES 传输相关的链路启动触发信号。

DSTC 启动由链路重启触发信号(C) 再次触发的 DES 传输。DSTC 再次执行 32768 次传输。32768 次传输后 DES 传输数量计数器读取 (1,1)。 DES 的 DV 为 01 时，DSTC 执行 DES 相关的 DES 关闭进程。第四个 DES 的 CHRS[5:4] 为 01 时，DSTC 设 SWTR:SWST 为 1，并结束传输。

传输结束后储存的 DES 值

传输操作示例 4 中的传输结束时，如 Table 4-7 所示更新 DES 值。表中黑体字表示与传输启动前不相同的值。在 DES2 和 DES3 中，第三个外循环中的传输启动地址已储存。

Table 4-7 传输操作示例 4 中传输结束后的 DES 值

DES 编号	值
DES0	DV = 00，ST = 00，其他值与启动传输时的值相同
DES1	ORM = 0x0001，IIN 与启动传输时相同。
DES2	SA = 0x00040000
DES3	DA = 0x00140000

补充信息

如上文所述，由于 CHRS[3:2] 设为 11，DSTC 可触发从当前 DES 的链路启动至再次触发当前 DES 的链路启动。通常，在 MODE 设为 0 的传输中，一个启动触发信号可触发执行 IIN 次传输。然而，如果 CHRS[3:2] 设为 11，一个启动触发信号可触发执行 ORM×IIN 次传输。

在传输操作示例 4 中，DSTC 将总共 98304 次传输划分为三部分，并单独执行这三部分 (ORM = 3) 的传输 (IIN = 32768)。只要 ORM 和 IIN 之积与传输总数量相同，不管将传输划分为多少部分，传输结果仍然相同。DSTC 一旦启动传输后，只能在进入启动触发等待状态或满足链路启动时间时才能启动另一传输的处理。因此，如传输操作示例 4 所述，当 DSTC 传输大量数据时，可能延迟启动另一个 HW 传输请求的传输。要避免发生延迟，需在满足 ORM 和 IIN 之积与传输总数量相同的条件下调整 ORM 值，这样可将大量的传输划分为更小的部分。因此，DSTC 传输大量数据时可以不延迟其他 HW 传输请求。

相反，要防止 DSTC 在链路启动时处理另一个 HW 传输请求，通过设 CHLK 为 1 使用链路锁定，这样 DSTC 可连续执行传输。

注意事项：

- *在 HW 传输中，如果 DES 传输由外设的启动请求所触发且每次数据传输时外设都必须与 DSTC 握手，则 CHRS 不能设为 11。*

4.5 传输操作示例 5

本节说明传输操作示例 5。

传输启动时的 DES 值

传输操作示例 5 的详细说明如下。

Table 4-8 传输操作示例 5 中传输启动时的 DES 值

区	DES 编号	值
1st-DES	DES0	DES0 = 0x612806B1 DV = 01：将在传输结束时执行 DES 关闭进程 MODE = 1，TW = 10：模式 1，32 位（字）传输 ORL = 101：外部重载 DES1 <= DES4，DES3 <= DES6 SAC = 001：递增 TW×1，需要内部重载 DAC = 000：递增 TW×1，无需内部重载 CHRS = 101000：在后续 DES 中有链路启动；未设置中断标志。 DMSET = 0：在 DES 关闭进程中未设置 DQMSK[n] CHLK = 0：无链路锁定 ACK = 01：通过硬件直接启动 DES 时，设 ACK 为 "01"。 PCHK = 0110：奇偶校验
	DES1	ORM = 0x0002，IIN = 0x02，IRM = 0x02
	DES2	SA = 0x00000000
	DES3	DA = 0x00001000
	DES4, 6	分别与第一个 DES 中的 DES1 和 DES3 相同
2nd-DES	DES0	DES0 = 0x60202063 DV = 11：在传输结束时不执行 DES 关闭进程 MODE = 0，TW = 00：模式 0，8 位（字节）传输 ORL = 011：外部重载 DES1 <= DES4，DES2 <= DES5 SAC = 000：递增 TW×1，无需内部重载 DAC = 001：具有内部重载时递增 TW×1,需要内部重载 CHRS = 100000：在后续 DES 中有链路启动；未设置中断标志。 DMSET = 0：在 DES 关闭进程中未设置 DQMSK[n] CHLK = 0：无链路锁定 ACK = 00：通过链路启动直接启动 DES 时，设 ACK 为 "00"。 PCHK = 0110：奇偶校验
	DES1	ORM = 0x0002，IIN = 0x0003
	DES2	SA = 0x00002000
	DES3	DA = 0x00003000
	DES4,6	分别与第二个 DES 中的 DES1 和 DES2 相同

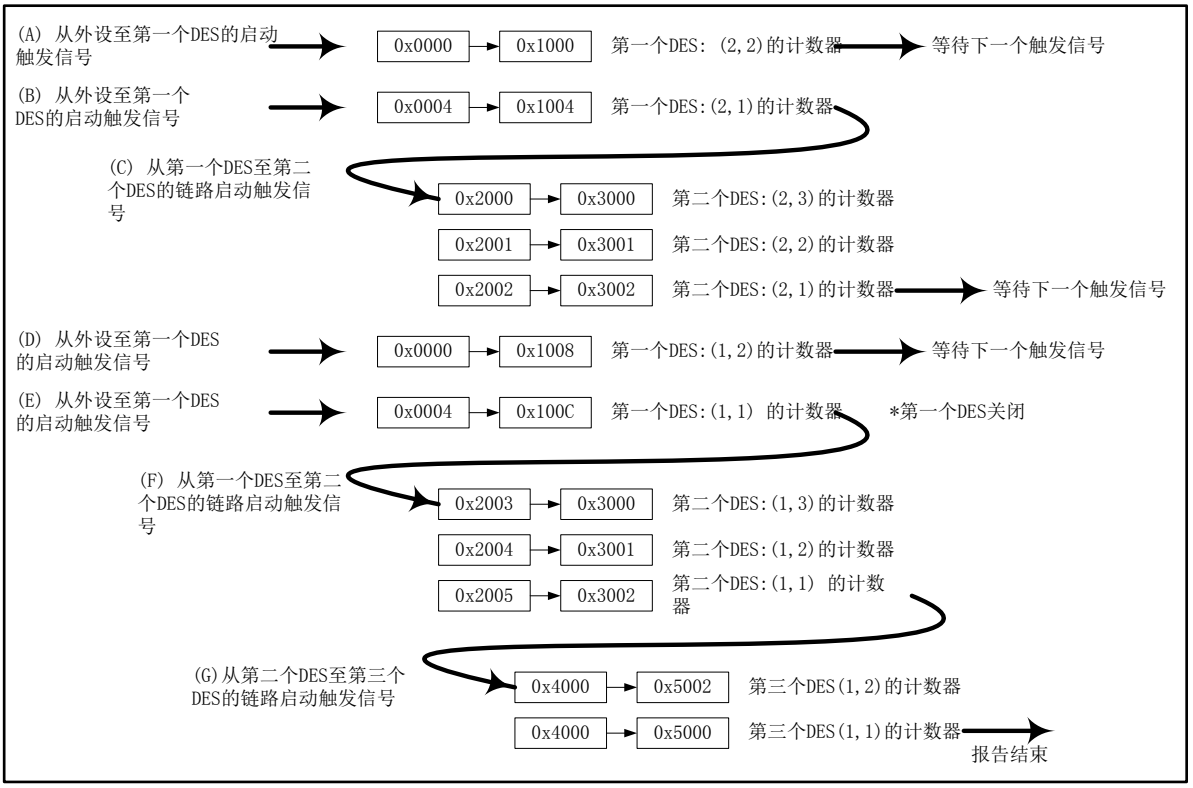
区	DES 编号	值
3rd-DES	DES0	DES0 = 0xB010F503 DV = 11：在传输结束时不执行 DES 关闭进程 MODE = 0，TW = 01：模式 0，16 位（半字）传输 ORL = 000：无需外部重载 SAC = 101：固定地址 DAC = 111：递减 TW×1，需要内部重载。 CHRS = 010000：无链路启动，已设置中断标志。 DMSET = 0：在 DES 关闭进程中未设置 DQMSK[n] CHLK = 0：无链路锁定 ACK = 00：通过链路启动直接启动 DES 时，设 ACK 为 "00"。 PCHK = 1011：奇偶校验
	DES1	ORM = 0x0001，IIN = 0x0002
	DES2	SA = 0x00004000
	DES3	DA = 0x00005002

本示例说明相对复杂的链路启动设置操作。在本示例中使用三个 DES。Table 4-8 分别列出第一个 DES、第二个 DES 和第三个 DES 的值。

传输操作流程

Figure 4-5 所示为传输操作示例 5 中的传输操作流程。Figure 4-5 中的启动触发信号 (A)，(B)，(D) 和 (E) 指 HW 启动传输以及与外设传输请求信号的对应。Figure 4-5 中的启动触发信号 (C)、(F) 和 (G) 为链路启动触发。

Figure 4-5 传输操作示例 5 操作流程



DSTC 启动由启动触发信号 (A) 所触发的第一个 DES 传输。 DSTC 执行一次 32 位传输至 0x0000 ~ 0x1000 地址区域。第一个 DES 传输数量计数器为 (2,2)。根据第一个 DES 的 CHRS[1:0] 设置 (CHRS[1:0] = 00)，DSTC 不将 HWINT[n] 寄存器设为 1。DSTC 等待下一个启动触发信号。

DSTC 启动由启动触发信号 (B) 再次触发的第一个 DES 传输。DSTC 执行一次 32 位传输至 0x0004 ~ 0x1004 地址区域。第一个 DES 传输数量计数器为 (2,1)。第一个 DES 的 CHRS[3:2] 为 10 时，发出后续第二个 DES 传输的链路启动触发信号。

DSTC 启动由启动触发信号 (C) 所触发的第二个 DES 传输。DSTC 启动 8 位传输至 0x2000 ~ 0x3000 地址区域。DSTC 相继执行 8 位传输三次 (IIN = 3) 。传输后第二个 DES 的传输数量计数器从 (2,3) 开始计数并读取 (2,1)。根据第二个 DES 的 CHRS[3:2] 设置 (CHRS[3:2] = 00)，DSTC 不将 HWINT[n] 寄存器设为 1。DSTC 等待下一个启动触发信号。

DSTC 启动由启动触发信号 (D) 再次触发的第一个 DES 传输。地址的内部重载适用于 SA。DA 保持增量。DSTC 执行一次 32 位传输至 0x0000 ~ 0x1008 地址区域。传输数量计数器读取 (1,2)。根据第一个 DES 的 CHRS[1:0] 设置 (CHRS[1:0] = 00)，DSTC 不将 HWINT[n] 寄存器设为 1。DSTC 等待下一个启动触发信号。

DSTC 启动由启动触发信号 (E) 再次触发的第一个 DES 传输。DSTC 执行一次 32 位传输至 0x0004 ~ 0x100C 地址区域。第一个 DES 的传输数量计数器为 (1,1)。第一个 DES 的 DV 为 01 时，DSTC 将执行第一个 DES 的 DES 关闭进程。第一个 DES 的 CHRS[5:4] 为 10 时，发出后续第二个 DES 传输的链路启动触发信号。(注意第一个 DES 中的 CHRS[5:0] 已设为 101000。)

DSTC 启动由链路启动触发信号 (F) 再次触发的第二个 DES 传输。 SA 保持增量。地址的内部重载适用于 DA。DSTC 相继执行 8 位传输三次 (IIN = 3) 。传输后第二个 DES 的传输数量计数器从 (1,3) 开始

计数并读取 (1,1)。第二个 DES 的 DV 为 11 时，DSTC 不执行第二个 DES 相关的 DES 关闭进程。第二个 DES 的 CHRS[5:4] 为 10 时，发出后续第三个 DES 传输相关的链路启动触发信号。

DSTC 启动由启动触发信号(G) 所触发的第三个 DES 传输。DSTC 相继执行 16 位传输两次 (IIN = 2)。传输后第三个 DES 的传输数量计数器从 (1,2) 开始计数,并读取 (1,1)。第三个 DES 的 DV 为 11 时，DSTC 不执行第三个 DES 相关的 DES 关闭进程。根据第三个 DES 的 CHRS[5:4] 设置 (CHRS[5:4] = 01)，DSTC 将 HWINT[n] 寄存器设为 1。

在传输操作示例 5 中，传输操作如上文所述结束。本示例中设置 DSTC 执行以下操作。只有在启动触发信号 (B) 和 (E) 所触发的传输操作后才能发出从第一个 DES 至第二个 DES 的链路启动触发信号。因此，在第三个 DES 的传输结束时将 HWINT[n] 寄存器设为 1。所有传输结束后，DSTC 通知 CPU 所有传输已结束。

传输结束后储存的 DES 值

传输操作示例 5 中的传输结束时，如 Table 4-9 所示更新 DES 值。表中黑体字表示与传输启动前不相同的值。由于 CPU 必须与 DSTC 握手，第一个 DES 设置为传输结束后由 DSTC 关闭。将第二个 DES 和第三个 DES 设置为不能由 DSTC 关闭。

Table 4-9 传输操作示例 5 中传输结束后的 DES 值

区	DES 编号	值
1st-DES	DES0	DV = 00，ST = 00，其他值与启动传输时的值相同
	DES1 到 DES4，DES6	与启动传输时相同
2nd-DES	DES0 到 DES5	与启动传输时相同
第三个 DES	DES0 到 DES3	与启动传输时相同

4.6 DSTC 控制示例

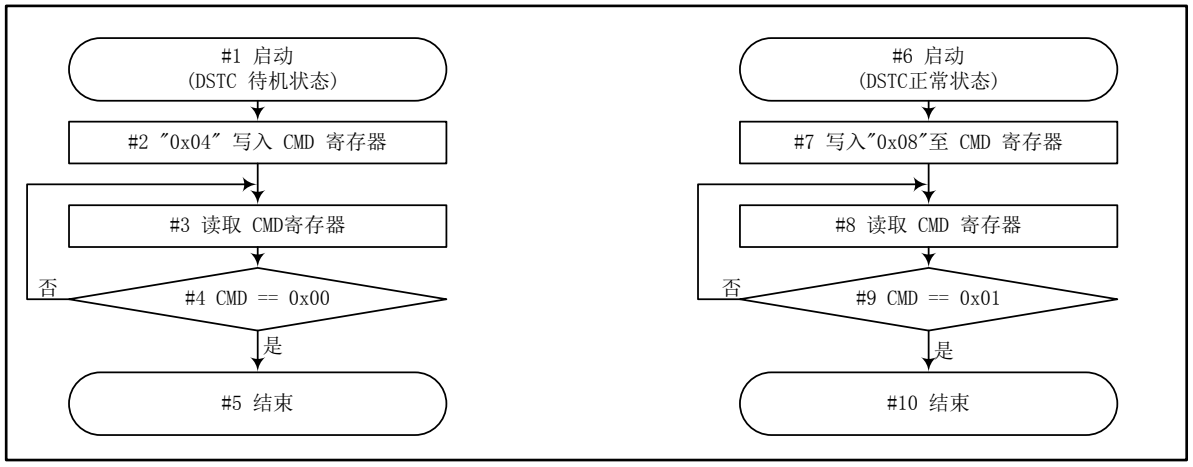
本节说明 DSTC 控制程序示例。

转换为待机状态和转换为正常状态的程序示例

DSTC 在总线复位后转换为待机状态。需要 DSTC 执行传输时，DSTC 首先要从待机状态转换至正常状态。如果不需要 DSTC 执行传输，可使 DSTC 处于待机状态以降低系统功耗。

Figure 4-6 所示为 DSTC 从待机状态转换至正常状态的程序示例以及 DSTC 从正常状态转换至待机状态的程序示例。本图中的编号与图后说明中使用的编号对应。

Figure 4-6 转换至待机状态和正常状态的示例



#1 ~ #5 所示为 DSTC 从待机状态转换至正常状态的程序。发出待机释放命令（CMD 寄存器写 0x04）。如果 DSTC 转换至正常状态，CMD 寄存器读为 0x00。检验 CMD 寄存器是否读为 0x00。

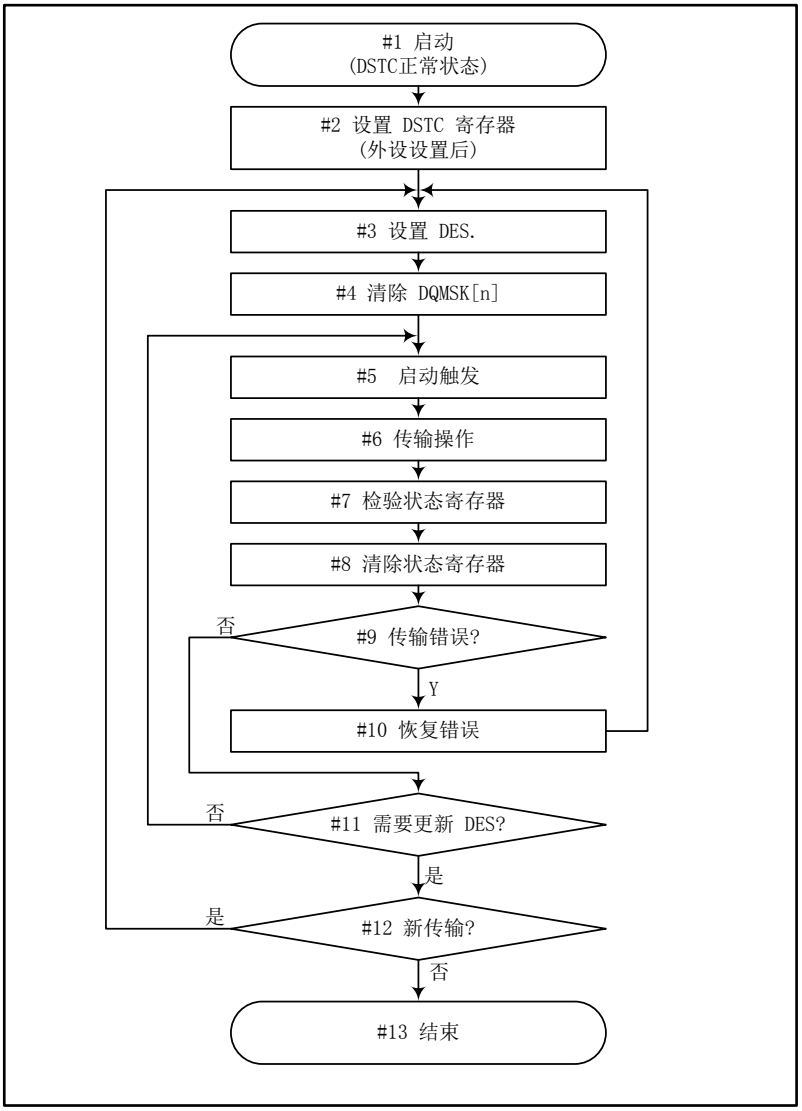
#6 ~ #10 所示为 DSTC 从正常状态转换至待机状态的程序。发出待机转换命令（ CMD 寄存器写 0x08）。如果 DSTC 转换至待机状态，CMD 寄存器将读为 0x01。检验 CMD 寄存器是否读为 0x01。

DSTC 执行传输时，如果发出待机转换命令，正在执行的传输将被强制结束，并执行该传输的 DES 关闭进程。完成 DES 关闭进程后，DSTC 转换至待机状态。

传输操作控制程序示例

Figure 4-7 说明控制 DSTC 传输操作的程序示例。Figure 4-7 中的编号与图后说明中使用的编号对应。

Figure 4-7 DSTC 传输操作控制程序示例



#1：本示例程序从 DSTC 处于正常状态这个点开始。如果 DSTC 未处于正常状态，不能执行以下进程。

#2： 说明 DSTC 的控制寄存器。分别将 DESTP 寄存器、CFG 寄存器、HWDESP[n] 寄存器和 DREQENB[n] 寄存器设为初始值。完成外设的设置后，将初始值写入 DREQENB[n] 寄存器。总线复位后清除 MONERS 寄存器。然而，DSTC 从待机状态再次解除后，先前因转换至待机状态而使传输强制结束的错误记录可能保留在 MONERS 寄存器中。使用 ERCLR 命令（CMD 寄存器写入 0x20），清除 MONERS 寄存器。

#3：在 CPU 中构建 DSTC 将使用的 DES 区域。

#4：总线复位后已清除 DQMSK[n] 寄存器。然而，可通过 DES 中 DMSET 所指定的 HW 传输错误、待机转换命令或源设置该寄存器。DQMSKCLR[n] 寄存器写入 1，清除 DQMSK[n] 寄存器。如果外设

的传输请求信号 (DREQ[n])已生效，CPU 对 DQMSK[n] 寄存器的清除将成为 #5 中的传输启动触发信号。

#5：发出启动触发信号。在 SW 传输中，通过 CPU 将 DESP 写入 SWTR 寄存器。有正在进行的 SW 传输时，不能写入 SWTR 寄存器。在 HW 传输中，外设的 DREQ[n] 信号生效即为启动触发信号。

#6：DSTC 根据 DES 执行传输操作。 DSTC 更新 DES 并恢复传输状态。

#7：通过 CPU 检查 DSTC 的传输状态。在 SW 传输中，读取 SWTR 寄存器的 SWREQ 位和 SWST 位。在 HW 传输中，读取 HWINT[n] 寄存器。读取 MONERS 寄存器，检查有无传输错误。

#8：通过 CPU 清除 DSTC 的传输状态。在 SW 启动传输中，通过 CPU 发出 SWCLR 命令（ CMD 寄存器写入 "0x10" ），清除 SWTR 寄存器的 SWST 位。在 HW 启动传输中，HWINTCLR[n] 寄存器写入 "1" ，清除 HWINT[n] 寄存器。

#9, #10：发生传输错误后，发出 ERCLR 命令（ CMD 寄存器写入 0x20 ），清除 MONERS 寄存器。传输因错误中断时，重建 DES。在 HW 传输中，需要设置外设。

#11 ~ #13：如果不需要重建 DES，发出下一个启动触发信号，启动传输。如果有需要重建 DES，通过 CPU 重建 DES，然后发出下一个启动触发信号，启动传输。

5. DSTC 寄存器和描述符

本节说明 DSTC 的寄存器功能和描述符功能。

5.1. 控制寄存器及 DES 列表
5.2. DESTP 寄存器
5.3. HWDESP[n] 寄存器
5.4. CMD 寄存器
5.5. CFG 寄存器
5.6.SWTR 寄存器
5.7. MONERS 寄存器
5.8. DREQENB[n] 寄存器
5.9. HWINT[n]寄存器
5.10. HWINTCLR[n] 寄存器
5.11. DQMSK[n] 寄存器
5.12. DQMSKCLR[n] 寄存器
5.13. 描述符 0 (DES0)
5.14. 描述符 1 (DES1)
5.15. 描述符 2 (DES2)
5.16. 描述符 3 (DES3)
5.17. 描述符 4 (DES4)
5.18. 描述符 5 (DES5)
5.19. 描述符 6 (DES6)

5.1 控制寄存器及 DES 列表

本节分别列出 DSTC 控制寄存器及 DES 列表。

Table 5-1 所示为 DSTC 控制寄存器列表，Table 5-2 为 DES 列表。

Table 5-1 DSTC 控制寄存器列表

地址	寄存器名称	参考章节
+0x00	DESTP	5.2
+0x04	HWDESP	5.3
+0x08	CMD	5.4
+0x09	CFG	5.5
+0x0A	SWTR	5.6
+0x0C	MONERS	5.7
+0x10 to +0x2F	DREQENB	5.8
+0x30 to +0x4F	HWINT	5.9
+0x50 to +0x6F	HWINTCLR	5.10
+0x70 to +0x8F	DQMSK	5.11
+0x90 to +0xAF	DQMSKCLR	5.12

Table 5-2 DES 列表

地址	描述符名称	参考章节
DESTP+DESP+0x00	DES0	5.13
DESTP+DESP+0x04	DES1	5.14
DESTP+DESP+0x08	DES2	5.15
DESTP+DESP+0x0C	DES3	5.16
DESTP+DESP+0x10 -	DES4	5.17
DESTP+DESP+0x10 -	DES5	5.18
DESTP+DESP+0x10 -	DES6	5.19

5.2 DESTP 寄存器

DESTP（描述符顶部地址）寄存器设置 DES 区域的启动地址。

寄存器配置

地址： +0x00

位	31	0
字段	DESTP[31:0]	
属性	R/W	
初始值	0x00000000	

寄存器功能

DSTC 访问并更新地址为 "DESTP + DESP" 的 DES。设置可读写内存区中的 DES 区域。DES 区域与字边界对齐。 DESTP 寄存器的低 2 位总是写入 "00" 。 DES 区域必须位于 DESTP 开始的 4096 字 (16 KB) 区域内。 DESTP 寄存器的设置值不能超过 "0xFFFFFFF0"。

在 DSTC 处于正常状态且正进行初始设置时设置 DESTP 寄存器。DSTC 执行传输时不能修改本寄存器值。修改本寄存器值后，DSTC 不能执行任何正常传输。

bit[31:0] DESTP（描述符顶部地址）

访问	功能
写入	写入这些位设置 DES 区域的启动地址。
读取	读取这些位的值。

5.3 HWDESP[n] 寄存器

HWDESP[n]（硬件 DES 指针）寄存器设置 HW 通道 n 发出传输请求时 DSTC 所访问 DES 的 DESP。

寄存器配置

地址： +0x04

位	31	30	29 ... 16	15 ... 8	7 ... 0
字段	保留		HWDESP[13:0]	保留	CHANNEL[7:0]
属性	R	R	R/W	R	R/W
初始值	0	0	XXXXXXXXXXXX00	00000000	00000000

寄存器功能

发出 HW 传输请求前设置 HWDESP[n] 寄存器。只有在 DSTC 处于正常状态时才能访问本寄存器。没必要设置不使用的 HW 通道。

HWDESP[n] 寄存器的编号对应 HW 通道的编号。但是通过 CPU 只能看到一个寄存器窗口。按照下文所述访问本寄存器。

- 通过 CPU 读取 HWDESP[n] 寄存器值时，首先使用 8 位（字节）写入要读取的通道编号 CHANNEL[7:0]。然后，以 16 位（半字）读取 HWDESP[13:0] 值。
- 要通过 CPU 写入 HWDESP[n] 寄存器值时，首先使用 8 位（字节）写入需写入的通道编号 CHANNEL[7:0]。然后，使用 16 位（半字）将值写入 HWDESP[13:0]。如果以 32 位（字）写入，可同时执行将值写入 CHANNEL[7:0] 和 HWDESP[13:0]。

如 Figure 3-5 所示，使用前，DSTC 将 HWDESP[n] 寄存器 DESP 值储存在 HWDESPBUF 中。如果相继发出通道 n 的 HW 启动请求，DSTC 将使用储存在 HWDESPBUF 中的 DESP 值而不是 HWDESP[n] 寄存器的 DESP 值。因此，通过 CPU 修改 HWDESP[n] 寄存器值时，储存在 HWDESPBUF 中的值将无效。修改 CFG 寄存器中的 RBDIS 位值可使 HWDESPBUF 的 DESP 值无效。有关 CFG 寄存器的详细说明，参见 “5.5CFG 寄存器”。

bit[7:0] CHANNEL[7:0]

访问	功能
写入	写入这些位，为将读取访问或写入访问的 HWDESP[n] 设置通道编号 (n)。
读取	读取这些位的值。

如果产品配备的 DSTC 支持 HW-128 通道，CHANNEL[7]始终写入 "0" 。

如果产品配备的 DSTC 支持 HW-64 通道，CHANNEL[7:6]始终写入 "00" 。

bit[29:16] HWDESP[13:0]

访问	功能
写入	写访问这些位，即写入 CHANNEL[7:0] 指定的 HWDESP[n] 通道编号。
读取	读取访问这些位，即读取 CHANNEL[7:0] 指定的 HWDESP[n] 通道编号。

DES 与字边界对齐。 HWDESP 寄存器的低 2 位始终写入 "00" 。 HWDESP 的设置值不能超过 "0x3FF0"。

5.4 CMD 寄存器

CMD（命令）寄存器向 DSTC 发出命令并读取 DSTC 的状态。

寄存器配置

地址： +0x08

位	7	6	5	4	3	2	1	0
字段	CMD[7:0]							
属性	W	W	W	W	W	W	R	R
初始值	0	0	0	0	0	0	0	1

寄存器功能

通过将值写入 CMD（命令）寄存器可以向 DSTC 发出命令。使用 8 位（字节）将值写入本寄存器。16 位和 32 位写入本寄存器将被忽略。此外，可读取本寄存器检验 DSTC 的状态（正常状态、待机状态、转换状态 1、和转换状态 2）。

在 DSTC 的各状态中，某些命令可以发出，某些命令不能发出。有关详细说明，参见 Table 3-13。

bit[7:0] CMD[7:0]

写入值	命令	进程参数
0x04	待机释放命令	指令 DSTC 从待机状态恢复至正常状态。
0x08	待机转换命令	指令 DSTC 从正常状态转换至待机状态。 将 SWTR:SWST 清除为 "0"。 使 SWINT 中断信号无效。 清除所有 HWINT[n] 寄存器。 使 HWINT[n] 中断信号无效。 使 ERINT 中断信号无效。 将所有 DQMSK[n] 寄存器设为 "1"。
0x10	SWCLR 命令	将 SWTR:SWST 清除为 "0"。 使 SWINT 中断信号无效。
0x20	ERCLR 命令	将 MONERS:EST 清除为 "0"。 使 ERINT 中断信号无效。 将 MONERS:DER 清除为 "0"。 将 MONERS:ESTOP 清除为 "0"。
0x80	MKCLR 命令	将所有 DQMSK[n] 寄存器清除为 "0"。 （转换状态 2 时忽略本命令。）
上述值以外的值	禁止将上述值以外的值写入 CMD[7:0]。（如果通过 RMW 访问方式等将上述值以外的值写入 CMD[7:0]，写入会被忽略。）	

读取值	DSTC 状态
0x00	指示 DSTC 处于正常状态。
0x01	指示 DSTC 处于待机状态。(初始值)
0x02	指示 DSTC 处于转换状态 1（从待机状态转换至正常状态）。
0x03	指示 DSTC 处于转换状态 2（从正常状态转换至待机状态）。
上述值以外的值	CMD 寄存器从不读取上述值以外的值。

5.5 CFG 寄存器

CFG（配置）寄存器设置 DSTC 的操作功能。

寄存器配置

地址： +0x09

位	15	14	13	12	11	10	9	8
字段	保留	SWPR[2:0]			ESTE	RBDIS	ERINTE	SWINTE
属性	R	R/W			R/W	R/W	R/W	R/W
初始值	0	100			0	0	0	0

寄存器功能

CFG（配置）寄存器设置 DSTC 的操作功能。通过 8 位（字节）访问将值写入本寄存器。16 位和 32 位写入本寄存器将被忽略。DSTC 处于正常状态时，不能写入本寄存器。

bit[8] SWINTE（软件中断使能）

访问	功能
写入 "0"	禁用 SWINT 中断。 (初始值) 如果 SWTR:SWST 已设为 "1"，DSTC 将不生成 SWINT 中断。
写入 "1"	使能 SWINT 中断。 如果 SWTR:SWST 已设为 "1"，DSTC 将生成 SWINT 中断。
读取	读取此位的值。

bit[9] ERINTE（错误中断使能）

访问	功能
写入 "0"	禁用 ERINT 中断。 (初始值) 如果 MONERS:EST 已设为 "001"，"010"，"100" 或 "101"，DSTC 将不生成 ERINT 中断。
写入 "1"	使能 ERINT 中断。 如果 MONERS:EST 已设为 "001"，"010"，"100" 或 "101"，DSTC 将生成 ERINT 中断。
读取	读取此位的值。

MONERS:EST=”011” 是待机转换命令所产生的传输强制停止错误。在这种情况下，ERINT 中断无效。有关详细说明，参见 “3.2.9 待机功能”。

bit[10] RBDIS（读取跳转缓冲禁用）

访问	功能
写入 "0"	使能 DES 的读取跳转缓冲功能。 (初始值) 使能 HWDESPBUF 的 HWDESP[n] 访问跳转功能。
写入 "1"	禁用读取跳转缓冲功能。 禁用 HWDESPBUF 的 HWDESP[n] 访问跳转功能。
读取	读取此位的值。

RBDIS 位设为 "0" 时，使能 Figure 3-5 所示的读取跳转缓冲功能和 HWDESPBUF 功能。因此，DSTC 跳过对内存上 DES 和 HWDESP[n] 寄存器的访问，也就加快了 DSTC 的处理速度。尽管如此，DSTC 可能不会通过 CPU 识别 DES0 中 DV 值是否改变或 HWDESP[n] 寄存器值是否修改。所以，如果 DES0 中的 DV 值已改变或 HWDESP[n] 寄存器值已修改，RBDIS 位写入 "1" 。如果 RBDIS 位设为 "1" ，

DSTC 不使用读取跳转缓冲功能或 HWDESPBUF 功能，而是进行内存上的 DES 值以及 HWDESP[n] 寄存器值的访问操作。 DSTC 使用更新的 DES 值和 HWDESP[n] 寄存器值执行进程后，可将 "0" 写入 RBDIS 寄存器，再次使能缓冲功能。

bit[11] ESTE（错误停止使能）

访问	功能
写入 "0"	即使发生传输错误，DSTC 也不进入错误停止状态。 (初始值) 出现另一个传输请求时，DSTC 启动请求的传输。
写入 "1"	发生传输错误时，DSTC 将进入错误停止状态。 出现另一个传输请求时，DSTC 将保留所请求传输的启动。
读取	读取此位的值。

bit[12:14] SWPR[2:0]（软件传输奇偶校验）

用判优器 2 判优时，如果 SW 传输请求与 HW 传输请求有冲突，判优器 2 将指定 SW 传输获得传输权利的概率。即使在 DSTC 执行传输时也可以修改 SWPR 位值。修改 CFG 寄存器 SWPR 位的值后，可应用于下一个 SW 启动触发信号。

访问	功能
写入 "000"	将 SW 传输的优先级设为最高优先级。 （如果正在进行 HW 传输时发出 SW 传输请求，将在 HW 传输结束后启动 SW 传输。）
写入 "001"	将 SW 传输获得传输权利的概率设为 1/2。
写入 "010"	将 SW 传输获得传输权利的概率设为 1/3。
写入 "011"	将 SW 传输获得传输权利的概率设为 1/7。
写入 "100"	将 SW 传输获得传输权利的概率设为 1/15。（初始值）
写入 "101"	将 SW 传输获得传输权利的概率设为 1/31。
写入 "110"	将 SW 传输获得传输权利的概率设为 1/63。
写入 "111"	将 SW 传输的优先级设为最低优先级。 （只有在无 HW 传输请求时才能启动 SW 传输。）
读取	读取这些位的值。

bit[15] 保留

写入访问此位时写入"0"。读取访问此位时读出 "0"。

5.6 SWTR 寄存器

SWTR（软件触发）寄存器发出 SW 启动传输的启动触发信号。

寄存器配置

地址： +0x0A

位	31	30	29	16
字段	SWST	SWREQ	SWDESP[13:0]	
属性	R	R	R/W	
初始值	0	0	00000000000000	

寄存器功能

写入访问本寄存器时，SWTR（软件触发）寄存器将发出 SW 传输的启动触发信号。以 16 位（半字）访问将值写入本寄存器。32 位（字）写入访问被忽略。如果已执行 SW 启动指令，且该传输未结束 (SWREQ ≠ 0) 或 DSTC 未处于正常状态 (CMD ≠ 00) 或 DSTC 处于错误停止状态 (ESTOP ≠ 0)，DSTC 将忽略 CPU 的写入并忽略新的 SW 启动传输请求。

bit[29:16] SWDESP[13:0]（软件 DES 指针）

写入将启动 DES 的 DESP 值。 DSTC 根据 DESTP+SWDESP 的 DES 区域传输数据。在 SW 传输进程中执行链路启动时，DSTC 将以链路启动所用 DESP 值更新 SWDESP。DES 与字边界对齐。SWDESP 寄存器的低 2 位始终写入 "00"。SWDESP 的设置值不能超过 "0x3FF0"。

访问	功能
写入	写入访问这些位，指定 SW 启动所启动的传输 DESP。
读取	这些位指示正在进行或已结束的 SW 启动传输的 DESP。

bit[30] SWREQ（软件请求）

SWREQ 位为只读位，指示 SW 传输是挂起状态，还是正在执行 SW 传输和链路启动传输。写入此位的值被忽略。写入访问（启动触发） SWTR 寄存器，设 SWREQ 位为 "1"。SW 传输正常结束、异常结束或等待启动触发时，SWREQ 复位为 "0"。

访问	功能
写入	写入此位的值没有意义。
读取 "0"	指示 SW 传输未请求或已结束。
读取 "1"	指示 SW 传输挂起或 DSTC 正在执行 SW 传输。

bit[31] SWST（软件状态）

SWST 位为只读位，将 SW 传输结束通知发送至 CPU。SW 传输的 DES 指定设置的中断标志。SW 传输正常结束时，设 SWST 为 "1"。通过 SWCLR 命令、待机转换命令或写入 SWTR 寄存器将 SWST 清除为 "0"。若 (CFG:SWINTE = 1)&&(SWTR:SWST == 1)，发出 SWINT 中断信号。

访问	功能
写入	写入此位的值没有意义。
读取 "0"	指示 SW 传输还未正常结束。
读取 "1"	指示 SW 传输已正常结束。

5.7 MONERS 寄存器

MONERS 寄存器显示已发生传输错误的参数。

寄存器配置

地址： +0x0C

位	31	30	29	28	27	26	25	24
字段	保留		EDESP[13:8]					
属性	R	R	R	R	R	R	R	R
初始值	0	0	X	X	X	X	X	X

位	23	22	21	20	19	18	17	16
字段	EDESP[7:0]							
属性	R	R	R	R	R	R	R	R
初始值	X	X	X	X	X	X	0	0

位	15	14	13	12	11	10	9	8
字段	ECH[7:0]							
属性	R	R	R	R	R	R	R	R
初始值	X	X	X	X	X	X	X	X

位	7	6	5	4	3	2	1	0
字段	保留	EHS	保留	ESTOP	DER	EST[2:0]		
属性	R	R	R	R	R	R		
初始值	X	X	X	0	0	000		

寄存器功能

MONERS 寄存器为只读寄存器。写入访问本寄存器被忽略。出现传输错误时，错误参数被记录在 MONERS 寄存器中。访问 MONERS 寄存器可以检验错误的参数。如果 MONER:EST 指示已发生错误，DSTC 可生成 ERINT 中断。 ERINTE 位设为 "1" 时，发出 NVIC 的 ERINT 中断信号。向 CMD 寄存器发出 ERCLR 命令可清除 MONERS 寄存器值和 ERINT 中断。

有关 MONERS 寄存器的详细说明，参见 “3.2.8 MONERS 寄存器”。

bit[2:0] EST[2:0]（错误状态）

EST 指示已发生错误的参数。若 EST ≠ 000，即使发生新的传输错误，DESP 仍将保留先前传输错误的参数。通过 ERCLR 命令将 ESTOP 清除为 "000"。

访问	功能
写入	DESP 写入值不导致操作的执行。
读取 "000"	未发生错误。(初始值)
读取 "001"	源访问错误
读取 "010"	目标访问错误
读取 "011"	待机转换命令造成传输强制停止错误
读取 "100"	DES 访问错误
读取 "101"	DES 打开错误
读取上述值以外的值	未定义

bit[3] DER（双重错误）

DER 位指示是否发生双重错误。将 EST[2:0] 位设为 "000" 以外的值并将 DER 位设为 "0" 时，如果出现新的传输错误，DER 位将设为 "1"。通过 ERCLR 命令将 DER 位清除为 "0"。

访问	功能
写入	DESP 写入值不导致操作的执行。
读取 "0"	指示未发生双重错误。(初始值)
读取 "1"	指示已发生双重错误。

bit[4] ESTOP（错误停止）

ESTOP 位指示 DSTC 处于错误停止状态。 CFG:ESTE 设为 "1" 时，如果发生传输错误，ESTOP 位设为 "1"。处于错误停止状态时，保留 DSTC 的传输启动。通过 ERCLR 命令将 ESTOP 位清除为 "0"。此位清除为 "0" 时，保留传输启动。此位为 "1" 时，SWTR 寄存器的写入被忽略。

访问	功能
写入	DESP 写入值不导致操作的执行。
读取 "0"	指示 DSTC 未处于错误停止状态。(初始值)
读取 "1"	指示 DSTC 处于错误停止状态。

bit[5] 保留

读取值不确定。写入此位的值被忽略。

bit[6] EHS（错误硬件软件）

EHS 位指示 HW 启动或 SW 启动已启动的 DES 是否已发生错误。若 EST ≠ 000，即使出现新的传输错误，DESP 仍将保留先前传输错误的参数。若 EST = 000，DESP 值未定义。

访问	功能
写入	这些位写入值不导致操作的执行。
读取 "0"	SW 启动或该 SW 启动的链路启动所启动的传输已发生错误。
读取 "1"	HW 启动或该 HW 启动的链路启动所启动的传输已发生错误。

bit[7] 保留

读取值不确定。写入此位的值被忽略。

bit[15:8] ECH[7:0]（错误硬件通道）

EST[2:0] 位不为 "000" 而 EHS 位为 "1" 时，ECH[7:0] 位指示发生错误的 HW 启动通道编号。若 EST ≠ 000，即使出现新的传输错误，ECH[7:0] 位仍将保留先前传输错误的 HW 启动通道编号。EST[2:0] 位为 "000" 或 EHS 位为 "0" 时，ECH[7:0] 位值不确定。

访问	功能
写入	这些位写入值不导致操作的执行。
读取	如果 HW 启动已启动的 DES 发生传输错误，ECH[7:0] 位指示 HW 通道的编号。

bit[29:16] EDESP [13:0]（错误 DES 指针）

EDESP[13:0] 位指示已造成传输错误的 DES 的 DESP。若 EST ≠ 000，即使出现新的传输错误，EDESP[13:0] 位仍将保留先前传输错误 DES 的 DESP。若 EST = 000，EDESP[13:0] 位值不确定。EDESP[1:0] 位读取值总为 "0"。

访问	功能
写入	这些位写入值不导致操作的执行。
读取	EDESP[13:0] 位指示已造成传输错误的 DES 的 DESP。

bit[31:30] 保留

读取值为 "00"。写入这些位的值被忽略。

5.8 DREQENB[n] 寄存器

DREQENB[n] 寄存器确定是否使用 HW 通道 n。

寄存器配置

地址		
0x10	字段	DREQENB[31:0]
+0x14	字段	DREQENB[63:32]
+0x18	字段	DREQENB[95:64]
+0x1C	字段	DREQENB[127:96]
+0x20	字段	DREQENB[159:128]
+0x24	字段	DREQENB[191:160]
+0x28	字段	DREQENB[223:192]
+0x2C	字段	DREQENB[255:224]
属性（适用于所有区域）		R/W
初始值（适用于所有区域）		0x00000000

寄存器功能

DREQENB[n]（DMA 请求使能）寄存器确定是否在初始设置中使用 HW 通道 n。DSTC 处于正常状态时，可写入本寄存器。

将 "1" 写入 DREQENB[n] 寄存器，使用 HW 通道 n。将 "0" 写入 DREQENB[n] 寄存器，不使用 HW 通道 n。DREQENB[n] 寄存器为 "0" 时，与 DSTC 连接的外设中断信号或传输请求信号 (DREQ[n]) 被忽略。 DSTC 不能修改 DREQENB[n] 寄存器的值。

若外设中断与发送至 DSTC 的传输请求复用，DREQENB[n] 寄存器值确定是选择外设中断信号还是选择 DSTC 的 HWINT[n] 用作连接 NVIC 的中断信号。有关详细说明，参见 “2 DSTC 操作概述及 DSTC 系统配置”。

bit[255:0] DREQENB[255:0]（DMA 请求使能）

访问	功能
写入 "0"	禁用外设的 DREQ 信号。(初始值)
写入 "1"	使能外设的 DREQ 信号。
读取	读取这些位的值。

如果产品的 DSTC 支持 HW-128 通道，DREQENB[255:128] 位为保留区，其值固定为 "0" 。

如果产品的 DSTC 支持 HW-64 通道，DREQENB[255:64] 位为保留区，其值固定为 "0" 。

5.9 HWINT[n] 寄存器

HWINT[n] 寄存器将 HW 传输结束通知发送至 CPU。

寄存器配置

地址		
+0x30	字段	HWINT[31:0]
+0x34	字段	HWINT[63:32]
+0x38	字段	HWINT[95:64]
+0x3C	字段	HWINT[127:96]
+0x40	字段	HWINT[159:128]
+0x44	字段	HWINT[191:160]
+0x48	字段	HWINT[223:192]
+0x4C	字段	HWINT[255:224]
属性 （适用于所有区域）		R
初始值 （适用于所有区域）		0x0000000000000000

寄存器功能

HWINT[n]（硬件传输中断）寄存器为只读寄存器，用于发送 HW 传输结束通知至 CPU。写入访问本寄存器被忽略。

如果通过 HW 启动方式启动 DES 后，HW 启动方式启动的 DES 或链路启动方式启动的 DES 中 CHRS 内指定有中断标志设置，则设 HWINT[n] 为 "1"。将 "1" 写入 HWINTCLR[n] 寄存器或发出待机转换命令可将 HWINT[n] 寄存器清除为 "0"。如果 HWINT[n] 寄存器设为 "1"，发出 NVIC 中断信号。

bit[255:0] HWINT[255:0]（硬件传输中断）

访问	功能
写入	不执行操作。
读取 "0"	指示已启动的 HW 传输还未正常结束。
读取 "1"	指示 HW 传输已正常结束。

如果产品的 DSTC 支持 HW-128 通道，HWINT[255:128] 位为保留区域，其值固定为 “0”。

如果产品配备的 DSTC 支持 HW-64 通道，HWINT[255:64] 位为保留区域，其值固定为 “0”。

5.10 HWINTCLR[n] 寄存器

HWINTCLR[n] 寄存器用于清除 HWINT[n] 寄存器。

寄存器配置

地址		
+0x50	字段	HWINTCLR[31:0]
+0x54	字段	HWINTCLR[63:32]
+0x58	字段	HWINTCLR[95:64]
+0x5C	字段	HWINTCLR[127:96]
+0x60	字段	HWINTCLR[159:128]
+0x64	字段	HWINTCLR[191:160]
+0x68	字段	HWINTCLR[223:192]
+0x6C	字段	HWINTCLR[255:224]
属性 （适用于所有区域）		W
初始值 （适用于所有区域）		0x0000000000000000

寄存器功能

HWINTCLR[n] 寄存器为只写寄存器，可通过 CPU 清除 HWINT[n] 寄存器。DSTC 未处于正常状态时，不能写入访问本寄存器。

将 "1" 写入本寄存器可将 HWINT[n] 寄存器清除为 "0"。本寄存器写入 "0" 被忽略。读取值总为 "0"。

bit[255:0] HWINTCLR[255:0]（硬件传输中断清除）

访问	功能
写入 "0"	不执行操作。
写入 "1"	将 HWINT[n] 寄存器清除为 "0"。
读取	本寄存器所有位的读取值总为 "0"。

如果产品的 DSTC 支持 HW-128 通道，HWINTCLR[255:128] 位为保留区域，其值固定为 “0”。

如果产品的 DSTC 支持 HW-64 通道，HWINTCLR[255:64] 位为保留区，其值固定为 "0"

5.11 DQMSK[n]寄存器

DQMSK[n] 寄存器指示 HW 启动传输请求是否受到抑制。

寄存器配置

地址		
+0x70	字段	DQMSK[31:0]
+0x74	字段	DQMSK[63:32]
+0x78	字段	DQMSK[95:64]
+0x7C	字段	DQMSK[127:96]
+0x80	字段	DQMSK[159:128]
+0x84	字段	DQMSK[191:160]
+0x88	字段	DQMSK[223:192]
+0x8C	字段	DQMSK[255:224]
属性 （适用于所有区域）		R
初始值 （适用于所有区域）		0x00000000

寄存器功能

DQMSK[n] 寄存器为只读寄存器。写入访问本寄存器被忽略。本寄存器为 "1" 时，指示发送至 DSTC 的 HW 启动传输请求 (DREQ[n]) 受到抑制。满足以下任意条件时，DSTC 将 DQMSK[n] 设为 "1" 并抑制传输请求。

- 传输时，HW 通道 n 发生传输错误。
- CPU 已向 CMD 寄存器发出待机转换命令。
- DES 中 HW 通道 n 传输的 DMSET 为 "1"，且 DSTC 已执行 DES 关闭进程。

满足以下任意条件时，DSTC 将 DQMSK[n] 清除为 "0" 并解除传输请求抑制。

- 已将 "1" 写入 DQMSKCLR[n] 寄存器。
- CPU 已向 CMD 寄存器发出待机转换命令。

bit[255:0] DQMSK[255:0]（DMA 请求屏蔽）

访问	功能
写入	不执行操作。
读取 "0"	指示未抑制外设发出的 DREQ[n] 信号。
读取 "1"	指示抑制外设的 DREQ[n] 信号。

如果产品的 DSTC 支持 HW-128 通道，DQMSK[255:128] 位为保留区域，其值固定为 “0”。

如果产品的 DSTC 支持 HW-64 通道，DQMSK[255:64] 位为保留区域，其值固定为 “0”。

5.12 DQMSKCLR[n] 寄存器

DQMSKCLR[n] 寄存器用于清除 DQMSK[n] 寄存器。

寄存器配置

地址		
+0x90	字段	DQMSKCLR[31:0]
+0x94	字段	DQMSKCLR[63:32]
+0x98	字段	DQMSKCLR[95:64]
+0x9C	字段	DQMSKCLR[127:96]
+0xA0	字段	DQMSKCLR[159:128]
+0xA4	字段	DQMSKCLR[191:160]
+0xA8	字段	DQMSKCLR[223:192]
+0xAC	字段	DQMSKCLR[255:224]
属性（适用于所有区域）		W
初始值（适用于所有区域）		0x00000000

寄存器功能

DQMSKCLR[n] 寄存器为只写寄存器。 DSTC 处于待机状态时，不能写入本寄存器。

将 "1" 写入本寄存器时，DQMSK[n] 寄存器将被清除为 "0"。清除 DQMSK[n] 寄存器将立即抑制 HW 传输启动。因此，先完成受抑制传输的外设设置和 DES 设置，再清除 DQMSK[n] 寄存器。

bit[255:0] DQMSKCLR[255:0]（DMA 请求屏蔽清除）

访问	功能
写入 "0"	不执行操作。
写入 "1"	将 DQMSK[n] 寄存器清除为 "0"。
读取	本寄存器所有位的读取值总为 "0"。

如果产品的 DSTC 支持 HW-128 通道，DQMSKCLR[255:128] 位为保留区域，其值固定为 “0”。

如果产品的 DSTC 支持 HW-64 通道，DQMSKCLR[255:64] 位为保留区域，其值固定为 “0”。

5.13 描述符 0 (DES0)

本节说明描述符 0 (DES0)的参数。 DES0 用于传输的基本设置。

描述符配置

地址： DESTP + DESP + 0x00

位	31	30	29	28	27	26	25	24
字段	PCHK[3:0]				Reserved		ACK[1:0]	
C 属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
D 属性	R	R	R	R	R	R	R	R

位	23	22	21	20	19	18	17	16
字段	CHLK	DMSET	CHRS[5:0]					
C 属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
D 属性	R	R	R	R	R	R	R	R

位	15	14	13	12	11	10	9	8
字段	DAC[1:0]			SAC[2:0]			TW	
C 属性	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
D 属性	R	R	R	R	R	R	R	R

位	7	6	5	4	3	2	1	0
字段	ORL[2:0]			MODE	ST[1:0]		DV[1:0]	
C 属性	R/W	R/W	R/W	R/W	R	R	R/W	R/W
D 属性	R	R	R	R	W	W	R/W	R/W

* DES 区域符号：

上表 C 行的属性指通过 CPU 访问的 R/W 属性。

上表 D 行的属性指通过 DSTC 访问的 R/W 属性。

DES 区域构建在内存中。由于所有初始值不确定，这里省略。

描述符功能

bit[1:0] DV[1:0]（描述符有效）

DV[1:0] 位指定 DES 所有权属于 CPU 还是 DSTC。DV[1:0] 位也指定是否在 DES 打开进程后执行传输以及是否执行 DES 关闭进程。在 DES 关闭进程中，DSTC 将 DV[1:0] 位更新为 "00"。

DV[1] 设为 "1" 时，增加重载传输计数器和传输地址的具体限制条件。如果 DV[1] 设为 "1" ，且 DES 的设置使 DES1、DES2 和 DES3 值不能恢复相应值，则 DSTC 通知 CPU 出现 DES 打开错误。

值	功能
00	所有权属于 CPU。不执行传输。不执行 DES 关闭进程。 （如果 DSTC 读取本值，DSTC 将通知 DES 打开错误。）
01	所有权属于 DSTC。执行传输。执行 DES 关闭进程。
10	所有权属于 DSTC。不执行传输。执行 DES 关闭进程。
11	所有权属于 DSTC。执行传输。不执行 DES 关闭进程。

bit[3:2] ST[1:0]（传输状态）

DES 指定的传输结束后，DSTC 在 DES 关闭进程中将结束状态值写入 ST[1:0] 位。如果 DSTC 不执行 DES 关闭进程，由 CPU 初始设置的值将保留在 ST[1:0] 位中。在启动传输时，本区域内的值对 DSTC 操作无效。

如果由于 DES 访问错误或 DES 打开错误异常结束传输，DSTC 既不执行 DES 关闭进程，也不在 ST[1:0] 位写入数据，因为 DSTC 不能访问 DES 区域。此类错误通知根据 DSTC 的 MONERS 寄存器执行，不是 DES 的 ST。

值	功能
00	传输正常结束。
01	传输源访问出现错误，传输异常结束。
10	传输目标访问出现错误，传输异常结束。
11	CPU 发出的待机转换命令强制性停止传输，传输异常结束。

bit[4] MODE（传输模式）

MODE 位选择传输模式。模式 0 时，一个启动触发信号触发 DSTC 执行数据传输 IIN 次。模式 1 时，一个启动触发信号触发 DSTC 执行一次数据传输。如果触发 DSTC 执行 HW 传输的外设启动触发信号需要每次传输数据都要与 DSTC 握手，则使用模式 1。DSTC 不修改本区域的值。

值	功能
0	在模式 0 时执行传输。
1	在模式 1 时执行传输。

bit[7:5] ORL[2:0]（外部重载）

ORL[2:0] 指定是否在 ORM×IIN 次传输结束后，是否外部重载 DES1 传输数量计数器 (ORM/IRM/IIN)、DES2 的传输源地址 (SA) 和 DES3 的传输目标地址 (DA)。DES 大小由 ORL 值确定。根据 ORL[2:0] 位的值，始于 DES4 ~ DES6 中 DESP 的相对地址各不相同。 DSTC 不修改本区域的值。

位	值	功能
ORL[0]	0	不执行 DES1 的外部重载。不需要 DES4 区域。
	1	执行 DES1 的外部重载。需要 DES4 区域。
ORL[1]	0	不执行 DES2 的外部重载。不需要 DES5 区域。
	1	执行 DES2 的外部重载。需要 DES5 区域。
ORL[2]	0	不执行 DES3 的外部重载。不需要 DES6 区域。
	1	执行 DES3 的外部重载。需要 DES6 区域。

bit[9:8] TW[1:0]（传输位宽）

TW[1:0] 指定单次传输的数据位宽。 DSTC 不修改本区域的值。

值	功能
00	8 位（字节）
01	16 位（半字）
10	32 位（字）
11	禁止设置（如果 DSTC 从 TW 读出 "11"，将通知出现 DES 打开错误。）

bit[12:10] SAC[2:0]（源地址控制）

SAC[2:0] 指定传输时更新传输源地址的方法。 DSTC 不修改本区域的值。设置为 DES0.DV[1]=1 且 DES2 需要重建（DES2 不恢复为启动值）时，DSTC 发出 DES 打开错误通知。

值	功能
000	每次传输时地址按 TW×1 增量，无需内部重载。
001	每次传输时地址按 TW×1 增量，需要内部重载。
010	每次传输时地址按 TW×2 增量，无需内部重载。
011	每次传输时地址按 TW×2 增量，需要内部重载。
100	每次传输时地址按 TW×4 增量，无需内部重载。
101	传输时传输地址保持不变。
110	每次传输时地址按 TW×1 减量，无需内部重载。
111	每次传输时地址按 TW×1 减量，需要内部重载

bit[15:13] DAC[2:0]（目标地址控制）

DAC[2:0] 指定传输时更新传输目标地址的方法。 DSTC 不修改本区域的值。设置为 DES0.DV[1]=1 且 DES3 需要重建（DES3 不恢复为启动值）时，DSTC 将发出 DES 打开错误通知。

值	功能
000	每次传输时地址按 TW×1 增量，无需内部重载。
001	每次传输时地址 TW×1 增量，需要内部重载。
010	每次传输时地址按 TW×2 增量，无需内部重载。
011	每次传输时地址按 TW×2 增量，需要内部重载。
100	每次传输时地址按 TW×4 增量，无需内部重载。
101	传输时传输地址保持不变。
110	每次传输时地址按 TW×1 减量，无需内部重载。
111	每次传输时地址按 TW×1 减量，需要内部重载。

bit[21:16] CHRS[5:0]（链路及唤醒状态）

CHRS[5:0] 位指定在执行指定次数的传输后将执行的进程。

位	选择条件	值	功能
CHRS[1:0]	IRM ≠ 1 ORM：忽略	00	不设置中断标志。无链路启动。
		01	中断标志已设置。无链路启动。
		10	不设置中断标志。在后续 DES 中有链路启动。
		11	不设置中断标志。在当前 DES 中有链路启动。
CHRS[3:2]	IRM = 1 ORM ≠ 1	00	不设置中断标志。无链路启动。
		01	中断标志已设置。无链路启动。
		10	不设置中断标志。在后续 DES 中有链路启动。
		11	不设置中断标志。在当前 DES 中有链路启动。
CHRS[5:4]	IRM = 1 ORM = 1	00	不设置中断标志。无链路启动。
		01	中断标志已设置。无链路启动。
		10	不设置中断标志。在后续 DES 中有链路启动。
		11	11: 禁止设置（发生 DES 打开错误。）

传输数量计数器的状态确定 DSTC 在当前 DES 后执行下一进程时应遵循 CHRS[5:4]、CHRS[3:2] 还是 CHRS[1:0]。如果有中断标志设置指令，SW 启动触发信号以及该 SW 启动触发信号之后的链路启动触发信号将 SWTR 寄存器的 SWST 位设为 1。HW 启动触发信号以及该 HW 启动触发信号之后的链路启动触发信号将 HWINT[n] 寄存器设为 1。当 MODE 为 0 时，因 CHRS[1:0] 无意义，写入 00。如果 MODE 为 0 而 CHRS[1:0] 不为 00，发生 DES 打开错误。DSTC 不修改本区域的值。

bit[22] DMSET（DREQ 屏蔽设置）

对于通过外设的通道 n 由 HW 启动方式直接启动的 DES，以及通过前述 DES 由链路启动所启动的 DES，DMSET 位设为 1 时，如果未执行 DES 关闭进程，则 DQMSK[n] 寄存器中与该 DES 对应的位设为 1。DSTC 不修改本区域的值。

值	功能
0	执行 HW 传输的 DES 关闭进程时，DQMSK[n] 寄存器不设为 1。
1	执行 HW 传输的 DES 关闭进程时，DQMSK[n] 寄存器设为 1。

bit[23] CHLK（链路锁定）

CHLK 位指定是否在当前传输后立即执行链路启动的下一个传输（链路锁定）或在链路启动的下一个传输前使能执行其他传输。 CHLK 位设为 1 时，如果 CHRS[5]、CHRS[3] 和 CHRS[1] 中任一项不为 1（选择的链路启动），则发生 DES 打开错误。DSTC 不修改本区域的值。

值	功能
0	当前传输后，可在链路启动传输前执行其他传输。
1	当前传输后立即执行链路启动传输。

bit[25:24] ACK[1:0]（确认）

ACK[1:0]设置的值用于调节使用 HW 传输时 DSTC 输出 DMA 传输请求确认信号至外围设备的时序。使用 HW 传输时，设置 ACK[1:0] 位为 "01"，通过外设设备的 HW 启动功能直接启动 DES。对于 HW 传输的链路启动方式启动的 DES、SW 传输中使用的 DES、以及 SW 传输的链路启动方式启动的 DES，则 ACK[1:0] 位设为 00。DSTC 不修改本区域的值。

值	功能
00	DSTC 输出 DMA 传输确认信号至连接 DSTC 的外设。
01	DSTC 输出 DMA 传输确认信号至连接 DSTC 的外设。
10, 11	保留

bit[27:26] 保留

本区域写入 00 。如果未将 00 写入本区域，DSTC 将通知 CPU 有 DES 打开错误。 DSTC 不修改本区域的值。

bit[31:28] PCHK[3:0]（奇偶校验）

PCHK[3:0] 位设置 DES0 区的奇偶校验（调用以下方程式）。

PCHK[3:0] != (DES0[27:24] ^ DES0[23:20] ^ DES0[19:16] ^ DES0[15:12] ^ DES0[11:8] ^ DES0[7:4])

构建 DES 时，CPU 将计算奇偶值。 DSTC 检验 PCHK[3:0] 与 DES0 区域值之间的一致性。若发生奇偶校验错误，DSTC 将通知 CPU 出现 DES 打开错误。DSTC 不修改本区域的值。PCHK[3:0] 的操作目标是 DSTC 不修改的 DES0[27:4] 区域。DES0[3:0] 值的改变对 PCHK[3:0] 值无效。

5.14 描述符 1 (DES1)

本节说明描述符 1 (DES1)的参数。 DES1 设置传输的数量。 DES1 区域在模式 0 (DES0.MODE=0) 和模式 1 (DES0.MODE=1) 中的配置和功能不同。

描述符配置（模式 0）

地址： DESTP + DESP + 0x04

位	31	16	15	0
字段	ORM[15:0]		IIN[15:0]	
C 属性	R/W		R/W	
D 属性	R/W		R/W	

描述符功能（模式 0）

bit[15:0] IIN[15:0]（内循环初值）

IIN[15:0] 位指定传输数量计数器中内循环计数器的初始值。可设置为 "1" 至 "65536" （含）范围之内的值。将 IIN[15:0] 位设为 "0x0000" 等效于设为 "65536"。DSTC 导入 IIN 值，作为内循环计数器剩余值 (IRM)。因此，在模式 0 中，有必要指定 IRM。

传输时 DSTC 不修改本区域的值。使能 DES1 的外部重载 (ORL[0] = 1) 时，DSTC 将 DES[15:0] 值复制至 IIN。

bit[31:16] ORM[15:0]（外循环余值）

ORM[15:0] 位指定传输数量计数器中外循环计数器的剩余值。可设置在 "1" 到 "65536" （含）范围之内的值。将 ORM[15:0] 位设为 "0x0000" 等效于设为 "65536"。

将 ORM 值写回 DES 前，DSTC 减少该值。传输结束时 DSTC 将 "0x0001" 储存在 ORM 中。使能 DES1 的外部重载时，在最终传输后 DSTC 将 DES4[31:16] 值复制到 ORM。传输因发生错误而结束时，DSTC 将储存传输结束前出现的值，于是需要通过 CPU 初始化 ORM[15:0] 位。

在模式 0 中，将传输数量设为以下范围内的值。若值超出范围，DSTC 将通知 CPU 出现 DES 打开错误。

若 ORM = 65536，则 IIN < 0x2000
若 ORM ≥ 0x8000，则 IIN < 0x4000
若 ORM ≥ 0x4000，则 IIN < 0x8000
若 ORM≥ 0x2000，则 IIN < 65536

（若 ORM 小于 0x2000，则 IIN 的值无限制。）

设置为 DES0.DV[1]=1 且 DES1 需要重建（DES1 不恢复为启动值）时，DSTC 发出 DES 打开错误通知。

描述符配置（模式 1）

地址： DESTP + DESP + 0x04

位	31	16	15	8	7	0
字段	ORM[15:0]		IRM[7:0]		IIN[7:0]	
C 属性	R/W		R/W		R/W	
D 属性	R/W		R/W		R/W	

描述符功能（模式 1）

bit[7:0] IIN[7:0]（内循环初始值）

IIN[15:0] 位指定传输数量计数器中内循环计数器的初始值。可设置为 "1" 至 "256" （含）范围之内的值。将 IIN[7:0] 位设为 "0x00" 等效于设为 "256"。

传输时 DSTC 不修改本区域的值。使能 DES1 的外部重载时，在最终传输后 DSTC 将 DES[7:0] 值复制至 IIN[7:0] 位。

bit[15:8] IRM[7:0]（内循环余值）

IRM[7:0] 位指定传输数量计数器中内循环计数器的剩余值。将 IRM[7:0] 位设为 IIN[7:0] 位相同的值。

将 IRM[7:0] 位值反写至 DES 前，DSTC 减少该值。传输结束时 DSTC 将 "0x01" 储存在 IRM 中。使能 DES1 的外部重载时，在最终传输后 DSTC 将 DES4[15:8] 值复制至 IRM[7:0] 位。传输因发生错误而结束时，DSTC 将储存在传输结束前出现的值，于是需要通过 CPU 初始化 IRM[7:0] 位。DSTC 检测到 IRM[7:0] 位值比 IIN[7:0] 位值大时，将通知系统出现 DES 打开错误。

bit[31:16] ORM[15:0]（外循环余值）

ORM[15:0] 位指示传输数量计数器中外循环计数器的剩余值。可设置在 "1" 到 "65536" （含）范围之内的值。将 ORM[15:0] 位设为 "0x0000" 等效于设为 "65536"。

将 ORM[15:0] 位值反写至 DES 前，DSTC 将减少该值。传输结束时 DSTC 将 "0x0001" 储存在 ORM[15:0] 位中。使能 DES1 的外部重载时，在最终传输后 DSTC 将 DES4[31:16] 值复制至 IRM[7:0] 位。传输因发生错误而结束时，DSTC 将储存在传输结束前出现的值，于是需要通过 CPU 初始化 ORM[15:0] 位。

设置为 DES0.DV[1]=1 且 DES1 需要重建（DES1 不恢复为启动值）时，DSTC 发出 DES 打开错误通知。

5.15 描述符 2 (DES2)

本节说明描述符 2 (DES2)的参数。

DES2 描述符配置

地址： DESTP + DESP + 0x08

位	31	0
字段	SA[31:0]	
C 属性	R/W	
D 属性	R/W	

DES2 描述符功能

bit[31:0] SA[31:0]（源地址）

SA[31:0] 位设置传输源地址。不能将 SA[31:0] 位设为与 DES0:TW 所指定的数据量不匹配的值。将 DES3 值反写至 DES 前，DSTC 将更新该值。使能 DES2 的外部重载 (ORL[1] = 1) 时，DSTC 将 DES5 值复制至 DES2。如果未使能传输源地址的内部重载/外部重载，注意传输结束时的值。传输因发生错误而结束时，DSTC 将储存在传输结束前出现的值，于是需要通过 CPU 初始化 SA[31:0] 位。

5.16 描述符 3 (DES3)

本节说明描述符 3 (DES3) 的参数。

DES3 描述符配置

地址： DESTP + DESP + 0x0C

位	31	0
字段	DA[31:0]	
C 属性	R/W	
D 属性	R/W	

DES3 描述符功能

bit[31:0] DA[31:0]（目标地址）

DA[31:0] 位设置传输目标地址。不能将 DA[31:0] 位设为与 DES0:TW 所指定的数据量不匹配的值。将 DES3 值反写至 DES 前，DSTC 将更新该值。使能 DES3 的外部重载 (ORL[2] = 1) 时，DSTC 将 DES6 值复制至 DES3。如果未使能传输目标地址的内部重载/外部重载，注意传输结束时的值。传输因发生错误而结束时，DSTC 将储存传输结束前出现的值，于是需要通过 CPU 初始化 DA[31:0] 位。

5.17 描述符 4 (DES4)

本节说明描述符 4 (DES4)的参数。

DES4 描述符配置

地址： DESTP + DESP + 0x10 (ORL[2:0] = xx1)

位	31	0
字段	DES4[31:0]	
C 属性	R/W	
D 属性	R	

DES4 描述符功能

bit[31:0] DES4[31:0]（描述符 4）

外部重载时 DES4 设置加载至 DES1 的值（传输数量）。在相同配置下将 DES4 设为 DES1 相同的值。DSTC 不修改本区域的值。

5.18 描述符 5 (DES5)

本节说明描述符 5 (DES5) 的参数。

DES5 描述符配置

地址： DESTP + DESP + 0x10 (ORL[2:0] = x10)

地址： DESTP + DESP + 0x14 (ORL[2:0] = x11)

位	31	0
字段	DES5[31:0]	
C 属性	R/W	
D 属性	R	

DES5 描述符功能

bit[31:0] DES5[31:0]（描述符 5）

外部重载时 DES5 设置将加载到 DES2 的设置（传输源启动地址）。将 DES5 设为与 DES2 相同的值。DSTC 不修改本区域的值。

5.19 描述符 6 (DES6)

本节说明描述符 6 (DES6)的参数。

DES6 描述符配置

地址： DESTP + DESP + 0x10 (ORL[2:0] = 100)

地址： DESTP + DESP + 0x14 (ORL[2:0] = 110, 101)

地址： DESTP + DESP + 0x18 (ORL[2:0] = 111)

位	31	0
字段	DES6[31:0]	
C 属性	R/W	
D 属性	R	

DES6 描述符功能

bit[31:0] DES6[31:0]（描述符 6）

外部重载时 DES6 设置加载至 DES3 的设置（传输源目标地址）。将 DES6 设为 DES3 相同的值。DSTC 不修改本区域的值。

附录

本章说明寄存器映射及注意事项列表。

A. 产品类型

B. 寄存器映射 (TYPE1-M0+)

C. 寄存器映射 (TYPE2-M0+)

D. 寄存器映射 (TYPE3-M0+)

E. 注意事项列表

代码：9AFAPPENDIXES-C03.0

A. 产品类型

本节说明产品类型。

1. 产品类型列表

代码：FM0_C3.0

1. 产品类型列表

本《手册》中，产品分类及描述如下。

有关 "TYPE1-M0+"、 “TYPE2-M0+” 和 “TYPE3-M0+” 等的描述，参见下表中 FM0+ 产品相关项：

Table 1-1 FM0+家族 TYPE1-M0+ 产品列表

类型	闪存大小	
	88K 字节	**56K 字节**
TYPE1-M0+	S6E1A12B0AGP20 S6E1A12B0AGN20 S6E1A12C0AGV20 S6E1A12C0AGF20 S6E1A12C0AGN20	S6E1A11B0AGP20 S6E1A11B0AGN20 S6E1A11C0AGV20 S6E1A11C0AGF20 S6E1A11C0AGN20

Table 1-2 FM0+家族 TYPE2-M0+产品列表

类型	闪存大小	
	560K 字节	**304K 字节**
TYPE2-M0+	S6E1B86G0AGV20 S6E1B86GHAGV20 S6E1B86F0AGV20 S6E1B86FHAGV20 S6E1B86E0AGV20 S6E1B86EHAGV20 S6E1B86E0AGF20 S6E1B86EHAGF20 S6E1B86E0AGB10 S6E1B86EHAGB10	S6E1B84G0AGV20 S6E1B84GHAGV20 S6E1B84F0AGV20 S6E1B84FHAGV20 S6E1B84E0AGV20 S6E1B84EHAGV20 S6E1B84E0AGF20 S6E1B84EHAGF20 S6E1B84E0AGB10 S6E1B84EHAGB10
	S6E1B36G0AGV20 S6E1B36F0AGV20 S6E1B36E0AGV20 S6E1B36E0AGF20 S6E1B36E0AGB10	S6E1B34G0AGV20 S6E1B34F0AGV20 S6E1B34E0AGV20 S6E1B34E0AGF20 S6E1B34E0AGB10

Table 1-3 FM0+家族 TYPE3 产品列表

类型	闪存大小	
	64K 字节	**128K** 字节
TYPE3-M0+	S6E1C31B0A S6E1C31C0A S6E1C31D0A S6E1C11B0A S6E1C11C0A S6E1C11D0A	S6E1C32B0A S6E1C32C0A S6E1C32D0A S6E1C12B0A S6E1C12C0A S6E1C12D0A

B. 寄存器映射(TYPE1-M0+)

本章说明寄存器映射图。

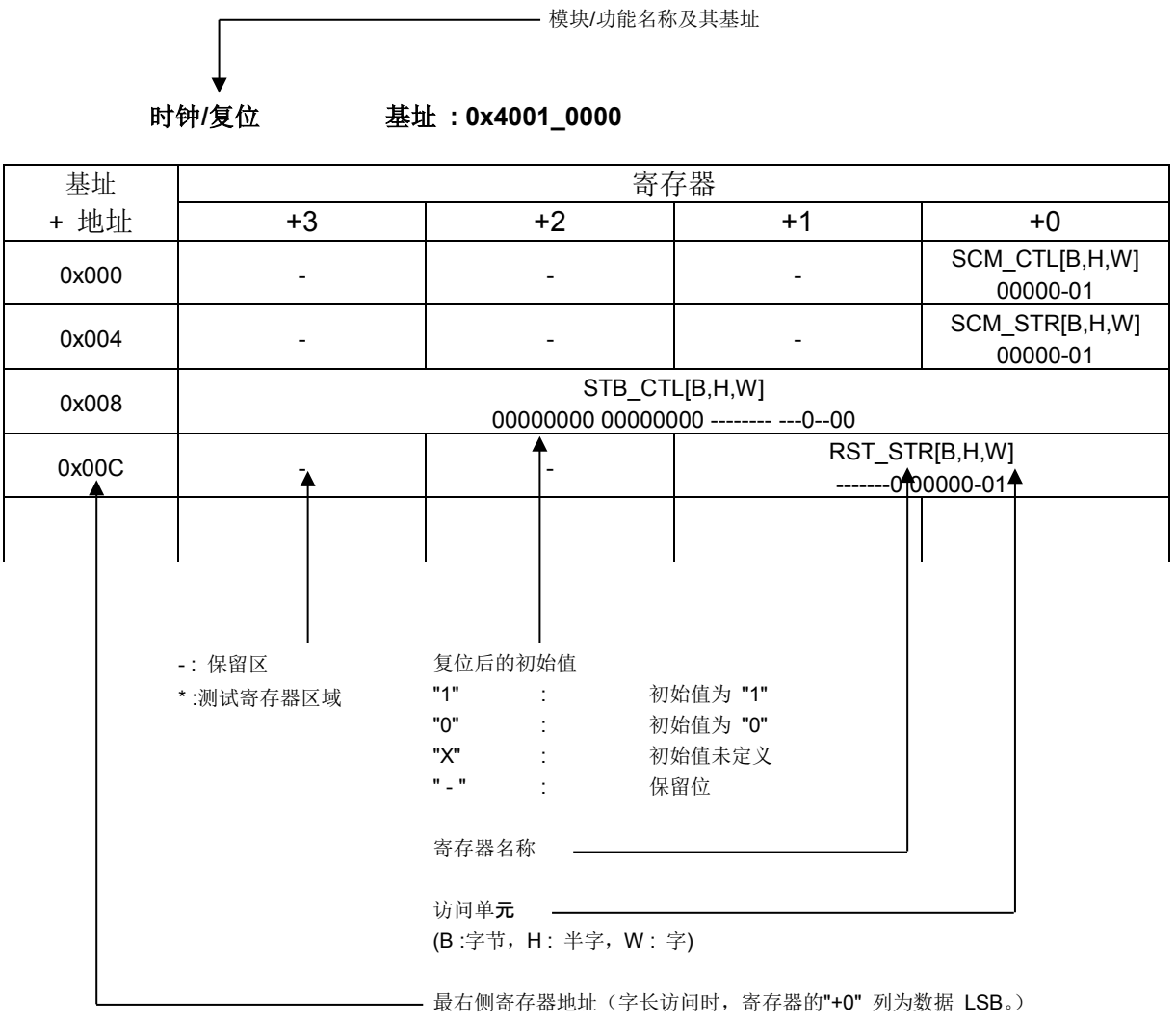
1. 寄存器映射

代码：9AFREGMAP-C01.0

1. 寄存器映射

以下为说明寄存器映射的模块/功能表。

[如何阅读各表]



注意事项：

- 寄存器表采用小端表示。
- 执行数据访问时，地址按访问量如下：
  - 字访问 ： 地址应为 4 的倍数（最低有效 2 位应为"0x00"）
  - 半字访问 ： 地址应为 2 的倍数（最低有效位应为"0x0"）
  - 字节访问 ： -
- 不可访问测试寄存器区。

- 不可访问寄存器表中未写入的区域。
- 如果寄存器的访问单元大于寄存器容量，同时也访问保留区，则读取值未定义，写入无效。

1.1 闪存 I/F

闪存 I/F 基址 ：0x4000_0000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	-	-
0x004	FRWTR[B,H,W]			
0x008	FSTR[B,H,W]			
0x00C				
0x010	FSYNDN[B,H,W]			
0x014 - 0x01C	-	-	-	-
0x020	FICR[B,H,W]			
0x024	FISR[B,H,W]			
0x028	FICLR[B,H,W]			
0x02C - 0x0FC	-	-	-	-
0x100	CRTRMM[B,H,W]			
0x104 - 0x1FC	-	-	-	-

注意事项：

- *有关闪存 I/F 寄存器的详细信息，参见所用产品的《闪存编程手册》。*

1.2 唯一 ID

唯一 ID 基址 ：0x4000_0200

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	UIDR0[W]			
	XXXXXXXX XXXXXXXXXXXXXXXX XXXX----			
0x004	UIDR1[W]			
	-------- -------- ---XXXXX XXXXXXXX			
0x008 – 0xDFC	-	-	-	-

1.3 时钟/复位

时钟/复位 基址 ：0x4001_0000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	-	SCM_CTL[W]
				00000-01
0x004	-	-	-	SCM_STR[W]
				00000-01
0x008	STB_CTL[W]			
	00000000 00000000 -------- ---0-000			
0x00C	-	-	RST_STR[W]	
			-------0 0000--01	
0x010	-	-	-	BSC_PSR[W]
				-----000
0x014	-	-	-	APBC0_PSR[W]
				------00
0x018	-	-	-	APBC1_PSR[W]
				1--0--00
0x01C	-	-	-	-
0x020	-	-	-	SWC_PSR[W]
				X-----00
0x024 - 0x02C	-	-	-	-
0x030	-	-	-	CSW_TMR[W]
				00000000
0x034	-	-	-	PSW_TMR[W]
				---0-000
0x038	-	-	-	PLL_CTL1[W]
				00000000
0x03C	-	-	-	PLL_CTL2[W]
				--000000
0x040	-	-	CSV_CTL[W]	
			-111--00 ------11	
0x044	-	-	-	CSV_STR[W]
				------00
0x048	-	-	FCSWH_CTL[W]	
			11111111 11111111	
0x04C	-	-	FCSWL_CTL[W]	
			00000000 00000000	
0x050	-	-	FCSWD_STR[W]	
			00000000 00000000	
0x054	-	-	-	DBWDT_CTL[W]
				0-0-----

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x058	-	-	-	*
0x05C	-	-	-	-
0x060	-	-	-	INT_ENR[W]
				--0--000
0x064	-	-	-	INT_STR[W]
				--0--000
0x068	-	-	-	INT_CLR[W]
				--0--000
0x06C - 0xFFC	-	-	-	-

1.4 HW WDT

HW WDT 基址 ：0x4001_1000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	WDG_LDR[W]			
	00000000 00000000 11111111 11111111			
0x004	WDG_VLR[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x008	-	-	-	WDG_CTL[W]
				------11
0x00C	-	-	-	WDG_ICL[W]
				XXXXXXXX
0x010	-	-	-	WDG_RIS[W]
				-------0
0x014 - 0xBFC	-	-	-	-
0xC00	WDG_LCK[W]			
	00000000 00000000 00000000 00000001			
0xC04 - 0xFFC	-	-	-	-

1.5 SW WDT

SW WDT 基址 ：0x4001_2000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	WdogLoad[W]			
	11111111 11111111 11111111 11111111			
0x004	WdogValue[W]			
	11111111 11111111 11111111 11111111			
0x008	-	-	-	WdogControl[W]
				---00000
0x00C	WdogIntClr[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x010	-	-	-	WdogRIS[W]
				-------0
0x014	*			
0x018	-	-	-	WdogSPMC[W]
				-------0
0x01C - 0xBFC	-	-	-	-
0xC00	WdogLock[W]			
	00000000 00000000 00000000 00000000			

0xC04 - 0xDFC	-	-	-	-
0xF00	*			
0xF08 - 0xFDF	-	-	-	-
0xFE0 - 0xFFC	*			

1.6 双计时器

双计时器 基址 ：0x4001_5000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	Timer1Load[W]			
	00000000 00000000 00000000 00000000			
0x004	Timer1Value[W]			
	11111111 11111111 11111111 11111111			
0x008	Timer1Control[W]			
	-------- -------- -------- 00100000			
0x00C	Timer1IntClr[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x010	Timer1RIS[W]			
	-------- -------- -------- -------0			
0x014	Timer1MIS[W]			
	-------- -------- -------- -------0			
0x018	Timer1BGLoad[W]			
	00000000 00000000 00000000 00000000			
0x020	Timer2Load[W]			
	00000000 00000000 00000000 00000000			
0x024	Timer2Value[W]			
	11111111 11111111 11111111 11111111			
0x028	Timer2Control[W]			
	-------- -------- -------- 00100000			
0x02C	Timer2IntClr[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x030	Timer2RIS[W]			
	-------- -------- -------- -------0			
0x034	Timer2MIS[W]			
	-------- -------- -------- -------0			
0x038	Timer2BGLoad[W]			
	00000000 00000000 00000000 00000000			
0x040 - 0xFFC	-	-	-	-

1.7 MFT

MFT 单元 0　　基址 ：0x4002_0000

MFT 单元 1　　基址 ：0x4002_1000

MFT 单元 2　　基址 ：0x4002_2000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x100	OCCP0[H,W]		-	-
	00000000 00000000			
0x104	OCCP1[H,W]		-	-
	00000000 00000000			
0x108	OCCP2[H,W]		-	-
	00000000 00000000			
0x10C	OCCP3[H,W]		-	-
	00000000 00000000			
0x110	OCCP4[H,W]		-	-
	00000000 00000000			
0x114	OCCP5[H,W]		-	-
	00000000 00000000			
0x118	-	OCSD10[B,H,W]	OCSB10[B,H,W]	OCSA10[B,H,W]
		00000000	00000000	00000000
0x11C	-	OCSD32[B,H,W]	OCSB32[B,H,W]	OCSA32[B,H,W]
		00000000	00000000	00000000
0x120	-	OCSD54[B,H,W]	OCSB54[B,H,W]	OCSA54[B,H,W]
		00000000	00000000	00000000
0x124	-	-	OCSC[B,H,W]	-
			--000000	
0x128	-	-	OCSE0[H,W]	
			00000000 00000000	
0x12C	OCSE1[H,W]			
	00000000 0000000000000000 00000000			
0x130	-	-	OCSE2[H,W]	
			00000000 00000000	
0x134	OCSE3[H,W]			
	00000000 0000000000000000 00000000			
0x138	-	-	OCSE4[H,W]	
			00000000 00000000	
0x13C	OCSE5[H,W]			
	00000000 0000000000000000 00000000			
0x140	TCCP0[H,W]		-	-
	1111111111111111			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x144	TCDT0[H,W]		-	-
	00000000 00000000			
0x148	TCSC0[B,H,W]		TCSA0[B,H,W]	
	00000000 00000000		000---00 01000000	
0x14C	TCCP1[H,W]		-	-
	1111111111111111			
0x150	TCDT1[H,W]			
	00000000 00000000			
0x154	TCSC1[B,H,W]		TCSA1[B,H,W]	
	00000000 00000000		000---00 01000000	
0x158	TCCP2[H,W]		-	-
	1111111111111111			
0x15C	TCDT2[H,W]		-	-
	00000000 00000000			
0x160	TCSC2[B,H,W]		TCSA2[B,H,W]	
	00000000 00000000		000---00 01000000	
0x164	TCAL[B,H,W] (only in unit 0)			
	00000000 00000000 1111111111111111			
0x168	-	OCFS54[B,H,W]	OCFS32[B,H,W]	OCFS10[B,H,W]
		00000000	00000000	00000000
0x16C	-	-	ICFS32[B,H,W]	ICFS10[B,H,W]
			00000000	00000000
0x170	-	ACFS54[B,H,W]	ACFS32[B,H,W]	ACFS10[B,H,W]
		00000000	00000000	00000000
0x174	ICCP0[H,W]		-	-
	0000000000000000			
0x178	ICCP1[H,W]		-	-
	0000000000000000			
0x17C	ICCP2[H,W]		-	-
	0000000000000000			
0x180	ICCP3[H,W]		-	-
	0000000000000000			
0x184	-	-	ICSB10[B,H,W]	ICSA10[B,H,W]
			------00	00000000
0x188	-	-	ICSB32[B,H,W]	ICSA32[B,H,W]
			------00	00000000
0x18C	WFTF10[H,W]		-	-
	0000000000000000			
0x190	WFTB10[H,W]		WFTA10[H,W]	
	0000000000000000		0000000000000000	
0x194	WFTF32[H,W]		-	-
	0000000000000000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x198	WFTB32[H,W]		WFTA32[H,W]	
	0000000000000000		0000000000000000	
0x19C	WFTF54[H,W]		-	-
	0000000000000000			
0x1A0	WFTB54[H,W]		WFTA54[H,W]	
	0000000000000000		0000000000000000	
0x1A4	-	-	WFSA10[H,W]	
			---00000 000000	
0x1A8	-	-	WFSA32[H,W]	
			---00000 000000	
0x1AC	-	-	WFSA54[H,W]	
			---00000 000000	
0x1B0	-	-	WFIR[H,W]	
			00000000 00000000	
0x1B4	-	-	NZCL[H,W]	
			-000--00 ---00000	
0x1B8	ACMP0		-	-
	00000000 00000000			
0x1BC	ACMP1		-	-
	00000000 00000000			
0x1C0	ACMP2		-	-
	00000000 00000000			
0x1C4	ACMP3		-	-
	00000000 00000000			
0x1C8	ACMP4		-	-
	00000000 00000000			
0x1CC	ACMP5		-	-
	00000000 00000000			
0x1D0	-	-	ACSA[B,H,W]	
			--000000 --000000	
0x1D4	-	-	ACSD0[B,H,W]	ACSC0[B,H,W]
			00000000	00000000
0x1D8	-	-	ACSD1[B,H,W]	ACSC1[B,H,W]
			00000000	00000000
0x1DC	-	-	ACSD2[B,H,W]	ACSC2[B,H,W]
			00000000	00000000
0x1E0	-	-	ACSD3[B,H,W]	ACSC3[B,H,W]
			00000000	00000000
0x1E4	-	-	ACSD4[B,H,W]	ACSC4[B,H,W]
			00000000	00000000
0x1E8	-	-	ACSD5[B,H,W]	ACSC5[B,H,W]
			00000000	00000000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x1EC - 0xFFC	-	-	-	-

1.8 PPG

PPG 基址 ：0x4002_4000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	TTCR0[B,H,W] 11110000	-
0x004	-	-	-	*
0x008	-	-	COMP0[B,H,W] 00000000	-
0x00C	-	-	-	COMP2[B,H,W] 00000000
0x010	-	-	COMP4[B,H,W] 00000000	-
0x014	-	-	-	COMP6[B,H,W] 00000000
0x018 - 0x01C	-	-	-	-
0x020	-	-	TTCR1[B,H,W] 11110000	-
0x024	-	-	-	*
0x028	-	-	COMP1[B,H,W] 00000000	-
0x02C	-	-	-	COMP3[B,H,W] 00000000
0x030	-	-	COMP5[B,H,W] 00000000	-
0x034	-	-	-	COMP7[B,H,W] 00000000
0x038 - 0x03C	-	-	-	-
0x040	-	-	TTCR2[B,H,W] 11110000	-
0x044	-	-	-	*
0x048	-	-	COMP8[B,H,W] 00000000	-
0x04C	-	-	-	COMP10[B,H,W] 00000000
0x050	-	-	COMP12[B,H,W] 00000000	-
0x054	-	-	-	COMP14[B,H,W] 00000000
0x058 - 0x0FC	-	-	-	-
0x100	-	-	TRG0[B,H,W] 00000000 00000000	

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x104	-	-	REVC0[B,H,W]	
			00000000 00000000	
0x108 - 0x13C	-	-	-	-
0x140	-	-	TRG1[B,H,W]	
			-------- 00000000	
0x144	-	-	REVC1[B,H,W]	
			-------- 00000000	
0x148 - 0x1FC	-	-	-	-
0x200	-	-	PPGC0[B,H,W]	PPGC1[B,H,W]
			00000000	00000000
0x204	-	-	PPGC2[B,H,W]	PPGC3[B,H,W]
			00000000	00000000
0x208	-	-	PRLH0[B,H,W]	PRLL0[B,H,W]
			XXXXXXXX	XXXXXXXX
0x20C	-	-	PRLH1[B,H,W]	PRLL1[B,H,W]
			XXXXXXXX	XXXXXXXX
0x210	-	-	PRLH2[B,H,W]	PRLL2[B,H,W]
			XXXXXXXX	XXXXXXXX
0x214	-	-	PRLH3[B,H,W]	PRLL3[B,H,W]
			XXXXXXXX	XXXXXXXX
0x218	-	-	-	GATEC0[B,H,W]
				--00---00
0x21C - 0x23C	-	-	-	-
0x240	-	-	PPGC4[B,H,W]	PPGC5[B,H,W]
			00000000	00000000
0x244	-	-	PPGC6[B,H,W]	PPGC7[B,H,W]
			00000000	00000000
0x248	-	-	PRLH4[B,H,W]	PRLL4[B.H.W]
			XXXXXXXX	XXXXXXXX
0x24C	-	-	PRLH5[B,H,W]	PRLL5[B,H,W]
			XXXXXXXX	XXXXXXXX
0x250	-	-	PRLH6[B,H,W]	PRLL6[B,H,W]
			XXXXXXXX	XXXXXXXX
0x254	-	-	PRLH7[B,H,W]	PRLL7[B,H,W]
			XXXXXXXX	XXXXXXXX
0x258	-	-	-	GATEC4[B,H,W]
				--00--00
0x25C - 0x27C	-	-	-	-
0x280	-	-	PPGC8[B,H,W]	PPGC9[B,H,W]
			00000000	00000000
0x284	-	-	PPGC10[B,H,W]	PPGC11[B,H,W]
			00000000	00000000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x288	-	-	PRLH8[B,H,W]	PRLL8[B,H,W]
			XXXXXXXX	XXXXXXXX
0x28C	-	-	PRLH9[B,H,W]	PRLL9[B,H,W]
			XXXXXXXX	XXXXXXXX
0x290	-	-	PRLH10[B,H,W]	PRLL10[B,H,W]
			XXXXXXXX	XXXXXXXX
0x294	-	-	PRLH11[B,H,W]	PRLL11[B,H,W]
			XXXXXXXX	XXXXXXXX
0x298	-	-	-	GATEC8[B,H,W]
				--00--00
0x29C - 0x2BC	-	-	-	-
0x2C0	-	-	PPGC12[B,H,W]	PPGC13[B,H,W]
			00000000	00000000
0x2C4	-	-	PPGC14[B,H,W]	PPGC15[B,H,W]
			00000000	00000000
0x2C8	-	-	PRLH12[B,H,W]	PRLL12[B,H,W]
			XXXXXXXX	XXXXXXXX
0x2CC	-	-	PRLH13[B,H,W]	PRLL13[B,H,W]
			XXXXXXXX	XXXXXXXX
0x2D0	-	-	PRLH14[B,H,W]	PRLL14[B,H,W]
			XXXXXXXX	XXXXXXXX
0x2D4	-	-	PRLH15[B,H,W]	PRLL15[B,H,W]
			XXXXXXXX	XXXXXXXX
0x2D8	-	-	-	GATEC12[B,H,W]
				--00--00
0x2DC - 0x2FC	-	-	-	-
0x300	-	-	PPGC16[B,H,W]	PPGC17[B,H,W]
			00000000	00000000
0x304	-	-	PPGC18[B,H,W]	PPGC19[B,H,W]
			00000000	00000000
0x308	-	-	PRLH16[B,H,W]	PRLL16[B,H,W]
			XXXXXXXX	XXXXXXXX
0x30C	-	-	PRLH17[B,H,W]	PRLL17[B,H,W]
			XXXXXXXX	XXXXXXXX
0x310	-	-	PRLH18[B,H,W]	PRLL18[B,H,W]
			XXXXXXXX	XXXXXXXX
0x314	-	-	PRLH19[B,H,W]	PRLL19[B,H,W]
			XXXXXXXX	XXXXXXXX
0x318	-	-	-	GATEC16[B,H,W]
				--00---00
0x31C - 0x33C	-	-	-	-
0x340	-	-	PPGC20[B,H,W]	PPGC21[B,H,W]
			00000000	00000000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x344	-	-	PPGC22[B,H,W]	PPGC23[B,H,W]
			00000000	00000000
0x348	-	-	PRLH20[B,H,W]	PRLL20[B,H,W]
			XXXXXXXX	XXXXXXXX
0x34C	-	-	PRLH21[B,H,W]	PRLL21[B,H,W]
			XXXXXXXX	XXXXXXXX
0x350	-	-	PRLH22[B,H,W]	PRLL22[B,H,W]
			XXXXXXXX	XXXXXXXX
0x354	-	-	PRLH23[B,H,W]	PRLL23[B,H,W]
			XXXXXXXX	XXXXXXXX
0x358	-	-	-	GATEC20[B,H,W]
				--00--00
0x35C - 0x37C	-	-	-	-
0x380	-	-	-	IGBTC[B,H,W]
				00000000
0x384 - 0xFFC	-	-	-	-

1.9 基本计时器

基本计时器 ch.0 基址： 0x4002_5000

基本计时器 ch.1 基址： 0x4002_5040

基本计时器 ch.2 基址： 0x4002_5080

基本计时器 ch.3 基址： 0x4002_50C0

基本计时器 ch.4 基址： 0x4002_5200

基本计时器 ch.5 基址： 0x4002_5240

基本计时器 ch.6 基址： 0x4002_5280

基本计时器 ch.7 基址： 0x4002_52C0

基本计时器 ch.8 基址： 0x4002_5400

基本计时器 ch.9 基址： 0x4002_5440

基本计时器 ch.10 基址： 0x4002_5480

基本计时器 ch.11 基址： 0x4002_54C0

基本计时器 ch.12 基址： 0x4002_5600

基本计时器 ch.13 基址： 0x4002_5640

基本计时器 ch.14 基址： 0x4002_5680

基本计时器 ch.15 基址： 0x4002_56C0

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	PCSR/PRLL[H,W]	
			XXXXXXXX XXXXXXXX	
0x004	-	-	PDUT/PRLH/DTBF[H,W]	
			XXXXXXXX XXXXXXXX	
0x008	-	-	TMR[H,W]	
			00000000 00000000	
0x00C	-	-	TMCR[B,H,W]	
			-0000000 00000000	
0x010	-	-	TMCR2[B,H,W]	STC[B,H,W]
			-------0	0000-000
0x014 - 0x03C	-	-	-	-

1.10 基本计时器的 IO 选择器

ch.0-ch.3（基本计时器）的 IO 选择器基址： 0x4002_5100

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	BTSEL0123[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

ch.4-ch.7（基本计时器）的 IO 选择器基址： 0x4002_5300

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	BTSEL4567[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

ch.8-ch.11（基本计时器）的 IO 选择器 基址： 0x4002_5500

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	BTSEL89AB[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

ch.12-ch.15（基本计时器）的 IO 选择器 基址： 0x4002_5700

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	BTSELCDEF[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

基于软件的同时启动（基本计时器） 基址： 0x4002_5F00

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000 - 0x0FB	-	-	-	-
0x0FC	-	-	BTSSSR[B,H,W]	
			XXXXXXXX XXXXXXXX	

1.11 QPRC

QPRC ch.0　　基址：　0x4002_6000

QPRC ch.1　　基址：　0x4002_6040

QPRC ch.2　　基址：　0x4002_6080

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x0000	-	-	QPCR[H,W]	
			00000000 00000000	
0x0004	-	-	QRCR[H,W]	
			00000000 00000000	
0x0008	-	-	QPCCR[H,W]	
			00000000 00000000	
0x000C	-	-	QPRCR[H,W]	
			00000000 00000000	
0x0010	-	-	QMPR[H,W]	
			11111111 11111111	
0x0014	-	-	QICRH[B,H,W]	QICRL[B,H,W]
			--000000	00000000
0x0018	-	-	QCRH[B,H,W]	QCRL[B,H,W]
			00000000	00000000
0x001C	-	-	QECR[B,H,W]	
			-------- -----000	
0x0020 - 0x0038	-	-	-	-
0x003C	QPCRR[B,H,W]		QRCRR[B,H,W]	
	00000000 00000000		00000000 00000000	

1.12 QPRC NF

QPRC ch.0 NF 基址： 0x4002_6100

QPRC ch.1 NF 基址： 0x4002_6110

QPRC ch.2 NF 基址： 0x4002_6120

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x0000	NFCTLA[B,H,W]			
	----------00-000			
0x0004	NFCTLB[B,H,W]			
	----------00-000			
0x0008	NFCTLC[B,H,W]			
	----------00-000			
0x000C	-	-	-	-

1.13 A/DC

12 位 A/DC 单元 0 基址 ：0x4002_7000

12 位 A/DC 单元 1 基址 ：0x4002_7100

12 位 A/DC 单元 2 基址 ：0x4002_7200

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	ADCR[B,H,W]	ADSR[B,H,W]
			000-0000	00---000
0x004	-	-	-	*
0x008	-	-	SCCR[B,H,W]	SFNS[B,H,W]
			1000-000	----0000
0x00C	SCFD[B,H,W]			
	XXXXXXXX XXXX---- ---X--XX ---XXXXX			
0x010	-	-	SCIS3[B,H,W]	SCIS2[B,H,W]
			00000000	00000000
0x014	-	-	SCIS1[B,H,W]	SCIS0[B,H,W]
			00000000	00000000
0x018	-	-	PCCR[B,H,W]	PFNS[B,H,W]
			1000-000	--XX--00
0x01C	PCFD[B,H,W]			
	XXXXXXXX XXXX---- ---X-XXX ---XXXXX			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x020	-	-	-	PCIS[B,H,W]
				00000000
0x024	CMPD[B,H,W]		-	CMPCR[B,H,W]
	00000000 00------			00000000
0x028	-	-	ADSS3[B,H,W]	ADSS2[B,H,W]
			00000000	00000000
0x02C	-	-	ADSS1[B,H,W]	ADSS0[B,H,W]
			00000000	00000000
0x030	-	-	ADST0[B,H,W]	ADST1[B,H,W]
			00010000	00010000
0x034	-	-	-	ADCT[B,H,W]
				00000111
0x038	-	-	SCTSL[B,H,W]	PRTSL[B,H,W]
			----0000	----0000
0x03C	-	-	ADCEN[B,H,W]	
			11111111 ------00	
0x040	*			
0x044	WCMRCIF[B,H,W]			
	-------- -------- -------- -------0			
0x048	WCMRCOT[B,H,W]			
	-------- -------- -------- -------0			
0x04C	-	-	WCMPSR[B,H,W]	WCMRPCR[B,H,W]
			--000000	001000--
0x050	WCMPDH[B,H,W]		WCMPDL[B,H,W]	
	0000000000------		0000000000------	
0x054 - 0x0FC	-	-	-	-

1.14 D/AC

10 位 D/AC 基址 ：0x4002_8000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x00	-	DACR0[B,H,W]	DADR0[B,H,W]	
		-------0	------XX XXXXXXXX	
0x04	-	DACR1[B,H,W]	DADR1[B,H,W]	
		-------0	------XX XXXXXXXX	
0x08 - 0xFC	-	-	-	-

1.15 CR 调节

CR 调节 基址 ：0x4002_E000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	-	MCR_PSR[B,H,W]
				------001
0x004	-	-	MCR_FTRM[B,H,W]	
			------01 11101111	
0x008	-	-	-	MCR_TTRM[B,H,W]
				---10000
0x00C	MCR_RLR[B,H,W]			
	00000000 00000000 00000000 00000001			
0x010 - 0x0FC	-	-	-	-

1.16 EXTI

EXTI 基址 ：0x4003_0000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	ENIR[B,H,W]			
	00000000 00000000 00000000 00000000			
0x004	EIRR[B,H,W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x008	EICL[B,H,W]			
	11111111 11111111 11111111 11111111			
0x00C	ELVR[R/W]			
	00000000 00000000 00000000 00000000			
0x010	ELVR1[R/W]			
	00000000 00000000 00000000 00000000			
0x014	-	-	NMIRR[B,H,W]	
			-------- -------0	
0x018	-	-	NMICL[B,H,W]	
			-------- -------1	
0x01C - 0x0FC	-	-	-	-

1.17 INT-Req. 读取

INT-Req. 读取　　基址 ：0x4003_1000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	DRQSEL[B,H,W]			
	00000000 00000000 00000000 00000000			
0x004	*			
0x008- 0x00B	-	-	-	-
0x00C	-	-	-	IRQCMODE[B,H,W]
				-------0
0x010	EXC02MON[B,H,W]			
	-------- -------- -------- ------00			
0x014	IRQ00MON[B,H,W]			
	-------- -------- -------- -------0			
0x018	IRQ01MON[B,H,W]			
	-------- -------- -------- -------0			
0x01C	IRQ02MON[B,H,W]			
	-------- -------- -------- -------0			
0x020	IRQ03MON[B,H,W]			
	-------- -------- ----0000 00000000			
0x024	IRQ04MON[B,H,W]			
	-------- -------- -------- 00000000			
0x028	IRQ05MON[B,H,W]			
	-------- 00000000 00000000 00000000			
0x02C	IRQ06MON[B,H,W]			
	-------- ----0000 00000000 00000000			
0x030	IRQ07MON[B,H,W]			
	-------- -------- -------- ------00			
0x034	IRQ08MON[B,H,W]			
	-------- -------- -------- ----0000			
0x038	IRQ09MON[B,H,W]			
	-------- -------- -------- ------00			
0x03C	IRQ10MON[B,H,W]			
	-------- -------- -------- ----0000			
0x040	IRQ11MON[B,H,W]			
	-------- -------- -------- ------00			
0x044	IRQ12MON[B,H,W]			
	-------- -------- -------- ----0000			
0x048	IRQ13MON[B,H,W]			
	-------- -------- -------- ------00			
0x04C	IRQ14MON[B,H,W]			
	-------- -------- -------- ----0000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x050	IRQ15MON[B,H,W]			
	-------- -------- -------- ------00			
0x054	IRQ16MON[B,H,W]			
	-------- -------- -------- ----0000			
0x058	IRQ17MON[B,H,W]			
	-------- -------- -------- ------00			
0x05C	IRQ18MON[B,H,W]			
	-------- -------- -------- ----0000			
0x060	IRQ19MON[B,H,W]			
	-------- -------- -------- ---0--00			
0x064	IRQ20MON[B,H,W]			
	-------- -------- -------- ---00000			
0x068	IRQ21MON[B,H,W]			
	-------- -------- -------- ---0--00			
0x06C	IRQ22MON[B,H,W]			
	-------- -------- -------- ---00000			
0x070	IRQ23MON[B,H,W]			
	-------- -------- -------0 00000000			
0x074	IRQ24MON[B,H,W]			
	-------- -------- -------- --00-000			
0x078	IRQ25MON[B,H,W]			
	-------- -------- -------- ---00000			
0x07C	IRQ26MON[B,H,W]			
	-------- -------- -------- ---00000			
0x080	IRQ27MON[B,H,W]			
	-------- -------- -------- --000000			
0x084	IRQ28MON[B,H,W]			
	-------- ------00 00000000 00000000			
0x088	IRQ29MON[B,H,W]			
	-------- -------- ----0000 00000000			
0x08C	IRQ30MON[B,H,W]			
	-------- ------00 00000000 00000000			
0x090	IRQ31MON[B,H,W]			
	----0--- -------- 00000000 00000000			
0x094 - 0x20C	-	-	-	-
0x210	RCINTSEL0[B,H,W] 00000000 00000000 00000000 00000000			
0x214	RCINTSEL1[B,H,W] 00000000 00000000 00000000 00000000			
0x218 - 0xFFC	-	-	-	-

1.18 LCDC

LCDC 基址 ：0x4003_2000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x00	-	LCDCC3[B,H,W]	LCDCC2[B,H,W]	LCDCC1[B,H,W]
		0011111-	--010100	-00000--
0x04	LCDC_PSR[B,H,W]			
	-------- -0000000 00000000 00000000			
0x08	LCDC_COMEN[B,H,W]			
	-------- -------- -------- 00000000			
0x0C	LCDC_SEGEN1[B,H,W]			
	00000000 00000000 00000000 00000000			
0x10	LCDC_SEGEN2[B,H,W]			
	-------- -------- -------- 00000000			
0x14	-	-	LCDC_BLINK[B,H,W]	
			00000000 00000000	
0x18	-	-	-	-
0x1C	LCDRAM03[B,H,W]	LCDRAM02[B,H,W]	LCDRAM01[B,H,W]	LCDRAM00[B,H,W]
	00000000	00000000	00000000	00000000
0x20	LCDRAM07[B,H,W]	LCDRAM06[B,H,W]	LCDRAM05[B,H,W]	LCDRAM04[B,H,W]
	00000000	00000000	00000000	00000000
0x24	LCDRAM11[B,H,W]	LCDRAM10[B,H,W]	LCDRAM09[B,H,W]	LCDRAM08[B,H,W]
	00000000	00000000	00000000	00000000
0x28	LCDRAM15[B,H,W]	LCDRAM14[B,H,W]	LCDRAM13[B,H,W]	LCDRAM12[B,H,W]
	00000000	00000000	00000000	00000000
0x2C	LCDRAM19[B,H,W]	LCDRAM18[B,H,W]	LCDRAM17[B,H,W]	LCDRAM16[B,H,W]
	00000000	00000000	00000000	00000000
0x30	LCDRAM23[B,H,W]	LCDRAM22[B,H,W]	LCDRAM21[B,H,W]	LCDRAM20[B,H,W]
	00000000	00000000	00000000	00000000
0x34	LCDRAM27[B,H,W]	LCDRAM26[B,H,W]	LCDRAM25[B,H,W]	LCDRAM24[B,H,W]
	00000000	00000000	00000000	00000000
0x38	LCDRAM31[B,H,W]	LCDRAM30[B,H,W]	LCDRAM29[B,H,W]	LCDRAM28[B,H,W]
	00000000	00000000	00000000	00000000
0x3C	LCDRAM35[B,H,W]	LCDRAM34[B,H,W]	LCDRAM33[B,H,W]	LCDRAM32[B,H,W]
	00000000	00000000	00000000	00000000
0x40	LCDRAM39[B,H,W]	LCDRAM38[B,H,W]	LCDRAM37[B,H,W]	LCDRAM36[B,H,W]
	00000000	00000000	00000000	00000000
0x44 - 0xFC	-	-	-	-

1.19 GPIO

GPIO 基址 ：0x4003_3000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	PFR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 1010			
0x004	PFR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x008	PFR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x00C	PFR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x010	PFR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x014	PFR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x018	PFR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x01C	PFR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x020	PFR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x024	PFR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x028	PFRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x02C	PFRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x030	PFRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x034	PFRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x038	PFRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x03C	PFRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x040 - 0x0FC	-	-	-	-
0x100	PCR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 1010			
0x104	PCR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x108	PCR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x10C	PCR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x110	PCR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x114	PCR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x118	PCR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x11C	PCR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x120	-	-	-	-
0x124	PCR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x128	PCRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x12C	PCRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x130	PCRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x134	PCRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x138	PCRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x13C	PCRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x140 - 0x1FC	-	-	-	-
0x200	DDR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x204	DDR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x208	DDR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x20C	DDR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x210	DDR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x214	DDR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x218	DDR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x21C	DDR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x220	DDR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x224	DDR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x228	DDRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x22C	DDRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x230	DDRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x234	DDRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x238	DDRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x23C	DDRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x240 - 0x2FC	-	-	-	-
0x300	PDIR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x304	PDIR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x308	PDIR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x30C	PDIR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x310	PDIR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x314	PDIR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x318	PDIR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x31C	PDIR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x320	PDIR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x324	PDIR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x328	PDIRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x32C	PDIRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x330	PDIRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x334	PDIRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x338	PDIRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x33C	PDIRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x340 - 0x3FC	-	-	-	-
0x400	PDOR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x404	PDOR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x408	PDOR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x40C	PDOR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x410	PDOR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x414	PDOR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x418	PDOR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x41C	PDOR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x420	PDOR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x424	PDOR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x428	PDORA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x42C	PDORB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x430	PDORC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x434	PDORD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x438	PDORE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x43C	PDORF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x440 - 0x4FC	-	-	-	-
0x500	ADE[B,H,W]			
	1111 1111 1111 1111 1111 1111 1111 1111			
0x504 - 0x57C	-	-	-	-
0x580	SPSR[B,H,W]			
	---- ---- ---- ---- ---- ---- ---- 0101			
0x584 - 0x5FC	-	-	-	-
0x600	EPFR00[B,H,W]			
	---- ---- ---- ---1 ---- ---- 0000 -000			
0x604	EPFR01[B,H,W]			
	0000 0000 0000 0000 --00 0000 0000 0000			
0x608	EPFR02[B,H,W]			
	0000 0000 0000 0000 ---0 0000 0000 0000			
0x60C	EPFR03[B,H,W]			
	0000 0000 0000 0000 ---0 0000 0000 0000			
0x610	EPFR04[B,H,W]			
	--00 0000 --00 00-- --00 0000 -000 00--			
0x614	EPFR05[B,H,W]			
	--00 0000 --00 00-- --00 0000 --00 00--			
0x618	EPFR06[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x61C	EPFR07[B,H,W]			
	---- 0000 0000 0000 0000 0000 0000 ----			
0x620	EPFR08[B,H,W]			
	---- 0000 0000 0000 0000 0000 0000 0000			
0x624	EPFR09[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x628 - 0x62C	-	-	-	-
0x630	EPFR12[B,H,W]			
	--00 0000 --00 00-- --00 0000 --00 00--			
0x634	EPFR13[B,H,W]			
	--00 0000 --00 00-- --00 0000 --00 00--			
0x638	EPFR14[B,H,W]			
	-------- ---- ---- ---- ------00 0000			
0x63C	EPFR15[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x640	EPFR16[B,H,W]			
	---- 0000 0000 0000 0000 0000 0000 ----			
0x644	EPFR17[B,H,W]			
	---- 0000 0000 0000 0000 0000 0000 ----			
0x648	EPFR18[B,H,W]			
	---- ---- ---- ---- ---- ---- ---- 0000			
0x64C - 0x650	-	-	-	-
0x654	EPFR21[B,H,W]			
	---- ---- ---- ---- ---- ---- ---- -000			
0x658	EPFR22[B,H,W]			
	---- ---- ---- ---- 0000 ---- 0000 ----			
0x65C - 0x6FC	-	-	-	-
0x700	PZR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x704	PZR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x708	PZR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x70C	PZR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x710	PZR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x714	PZR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x718	PZR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x71C	PZR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x720	PZR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x724	PZR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x728	PZRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x72C	PZRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x730	PZRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x734	PZRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x738	PZRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x73C	PZRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x740 - 0x7FC	-	-	-	-
0x800	*			
0x804	*			
0x808 - 0x8FC	-	-	-	-
0x900	FPOER0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x904	FPOER1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x908	FPOER2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x90C	FPOER3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x910	FPOER4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x914	FPOER5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x918	FPOER6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x91C	FPOER7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x920	FPOER8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x924	FPOER9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x928	FPOERA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x92C	FPOERB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x930	FPOERC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x934	FPOERD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x938	FPOERE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x93C	FPOERF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x940 - 0xFFC	-	-	-	-

1.20 HDMI-CEC

HDMI-CEC/远控接收器 ch.0 基址 ：0x4003_4000

HDMI-CEC/远控接收器 ch.1 基址 ：0x4003_4100

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x00	-	-	-	TXCTRL[B,H,W]
				--0000-0
0x04	-	-	-	TXDATA[B,H,W]
				00000000
0x08	-	-	-	TXSTS[B,H,W]
				--00---0
0x0C	-	-	-	SFREE[B,H,W]
				----0000
0x10 - 0x3F	-	-	-	-
0x40	-	-	RCCR[B,H,W]	RCST[B,H,W]
			0---0000	00000000
0x44	-	-	RCSHW[B,H,W]	RCDAHW[B,H,W]
			00000000	00000000
0x48	-	-	RCDBHW[B,H,W]	-
			00000000	
0x4C	-	-	RCADR1[B,H,W]	RCADR2[B,H,W]
			---00000	---00000
0x50	-	-	RCDTHH[B,H,W]	RCDTHL[B,H,W]
			00000000	00000000
0x54	-	-	RCDTLH[B,H,W]	RCDTLL[B,H,W]
			00000000	00000000
0x58	-	-	RCCKD[H,W]	
			---00000 00000000	
0x5C	-	-	RCRC[B,H,W]	RCRHW[B,H,W]
			---0---0	00000000
0x60	-	-	RCLE[B,H,W]	-
			00000-00	
0x64	-	-	RCLELW[B,H,W]	RCLESW[B,H,W]
			00000000	00000000
0x68 - 0xFC	-	-	-	-

1.21 LVD

LVD 基址 ：0x4003_5000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	LVD_CTL[B,H,W]	
			100000-- 000011--	
0x004	-	-	-	LVD_STR[B,H,W]
				0-------
0x008	-	-	-	LVD_CLR[B,H,W]
				1-------
0x00C	LVD_RLR[W]			
	00000000 00000000 00000000 00000001			
0x010	-	-	-	LVD_STR2
				01------
0x014 - 0x0FC	-	-	-	-

1.22 DS 模式

DS 模式 基址 ：0x4003_5100

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	-	REG_CTL[B,H,W]
				-------0
0x004	-	-	-	RCK_CTL[B,H,W]
				------01
0x008 - 0x6FC	-	-	-	-
0x700	-	-	-	PMD_CTL[B,H,W]
				-------0
0x704	-	-	-	WRFSR[B,H,W]
				------00
0x708	-	-	WIFSR[B,H,W]	
			------00 00000000	
0x70C	-	-	WIER[B,H,W]	
			------00 00000-00	
0x710	-	-	-	WILVR[B,H,W]
				-----000
0x714	-	-	-	DSRAMR[B,H,W]
				------00
0x718 - 0x7FC	-	-	-	-
0x800	BUR04[B,H,W]	BUR03[B,H,W]	BUR02[B,H,W]	BUR01[B,H,W]
	00000000	00000000	00000000	00000000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x804	BUR08[B,H,W]	BUR07[B,H,W]	BUR06[B,H,W]	BUR05[B,H,W]
	00000000	00000000	00000000	00000000
0x808	BUR012[B,H,W]	BUR11[B,H,W]	BUR10[B,H,W]	BUR09[B,H,W]
	00000000	00000000	00000000	00000000
0x80C	BUR16[B,H,W]	BUR15[B,H,W]	BUR14[B,H,W]	BUR13[B,H,W]
	00000000	00000000	00000000	00000000
0x810 - 0xEFC	-	-	-	-

1.23 MFS

MFS ch.0 基址 : 0x4003_8000

MFS ch.1 基址 : 0x4003_8100

MFS ch.2 基址 : 0x4003_8200

MFS ch.3 基址 : 0x4003_8300

MFS ch.4 基址 : 0x4003_8400

MFS ch.5 基址 : 0x4003_8500

MFS ch.6 基址 : 0x4003_8600

MFS ch.7 基址 : 0x4003_8700

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	SCR/IBCR[B,H,W]	SMR[B,H,W]
			0--00000	000-00-0
0x004	-	-	SSR[B,H,W]	ESCR/IBSR[B,H,W]
			0-000011	00000000
0x008	-	-	RDR/TDR[H,W]	
			00000000 00000000	
0x00C	-	-	BGR1[B,H,W]	BGR0[B,H,W]
			00000000	00000000
0x010	-	-	ISMK[B,H,W]	ISBA[B,H,W]
			--------	--------
0x014	-	-	FCR1[B,H,W]	FCR0[B,H,W]
			---00100	-0000000
0x018	-	-	FBYTE2[B,H,W]	FBYTE1[B,H,W]
			00000000	00000000
0x01C	-	-	SCSTR1/ EIBCR[B,H,W]	SCSTR0[B,H,W]
			00000000	00000000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x020	-	-	SCSTR3[B,H,W]	SCSTR2[B,H,W]
			00000000	00000000
0x024	-	-	SACSR[B,H,W]	
			--000--0 00-00000	
0x028	-	-	STMR[B,H,W]	
			00000000 00000000	
0x02C	-	-	STMCR[B,H,W]	
			00000000 00000000	
0x030	-	-	SCSCR[B,H,W]	
			00000000 00100000	
0x034	-	-	SCSFR1[B,H,W] 10000000	SCSFR0[B,H,W] 10000000
0x038	-	-	-	SCSFR2[B,H,W] 10000000
0x03C	-	-	TBYTE1[B,H,W]	TBYTE0[B,H,W]
			00000000	00000000
0x040	-	-	TBYTE3[B,H,W] 00000000	TBYTE2[B,H,W] 00000000
0x044 - 0x0FC	-	-	-	-

1.24 CRC

CRC 基址 ：0x4003_9000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	-	CRCCR[B,H,W]
				-0000000
0x004	CRCINIT[B,H,W]			
	11111111 11111111 11111111 11111111			
0x008	CRCIN[B,H,W]			
	00000000 00000000 00000000 00000000			
0x00C	CRCR[B,H,W]			
	11111111 11111111 11111111 11111111			

1.25 计时计数器

计时计数器 基址 ：0x4003_A000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	WCCR[B,H,W]	WCRL[B,H,W]	WCRD[B,H,W]
		00--0000	--000000	--000000

0x004 - 0x00C	-	-	-	-
0x010	-	-	CLK_SEL[B,H,W]	
			-----000 ------00	
0x014	-	-	-	CLK_EN[B,H,W]
				------00
0x018 - 0xFFC	-	-	-	-

1.26 RTC

RTC 基址 ：0x4003_B000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	WTCR1[B,H,W]			
	00000000 00000000 ---00000 -00000-0			
0x004	WTCR2[B,H,W]			
	-------- -------- -----000 -------0			
0x008	WTBR[B,H,W]			
	-------- 00000000 00000000 00000000			
0x00C	WTDR[B,H,W]	WTHR[B,H,W]	WTMIR[B,H,W]	WTSR[B,H,W]
	--000000	--000000	-0000000	-0000000
0x010	-	WTYR[B,H,W]	WTMOR[B,H,W]	WTDW[B,H,W]
		00000000	---00000	-----000
0x014	ALDR[B,H,W]	ALHR[B,H,W]	ALMIR[B,H,W]	-
	--000000	--000000	-0000000	
0x018	-	ALYR[B,H,W]	ALMOR[B,H,W]	-
		00000000	---00000	
0x01C	WTTR[B,H,W]			
	-------- ------00 0000000000000000			
0x020	-	-	WTCLKM[B,H,W]	WTCLKS[B,H,W]
			------00	-------0
0x024	-	WTCALEN[B,H,W]	WTCAL[B,H,W]	
		-------0	------00 00000000	
0x028	-	-	WTDIVEN[B,H,W]	WTDIV[B,H,W]
			------00	----0000
0x02C	-	-	-	WTCALPRD[B,H,W]
				--010011
0x030	-	-	-	WTCOSEL[B,H,W]
				-------0
0x034 - 0xFFC	-	-	-	-

1.27 低速 CR 预分频器

低速 CR 预分频器 基址 ：0x4003_C000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	-	LCR_PRSLD[B,H,W]
				--000000
0x000- 0x0FC	-	-	-	-

1.28 外设时钟选通

外设时钟选通 基址 ：0x4003_C100

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	CKEN0[B,H,W]			
	---1---1 ----1111 11111111 11111111			
0x004	MRST0[B,H,W]			
	-------0 ----00000000000000000000			
0x008 - 0x00C	-	-	-	-
0x010	CKEN1[B,H,W]			
	-------- ----1111 ----1111 ----1111			
0x014	MRST1[B,H,W]			
	-------- ----0000----0000----0000			
0x018 - 0x01C	-	-	-	-
0x020	CKEN2[B,H,W]			
	-------- -------- -------- --**---- Products with CAN : *="1" Products without CAN : *="0"			
0x024	MRST2[B,H,W]			
	-------- ----------00----			
0x028 - 0x0FC	-	-	-	-

1.29 DMAC

DMAC 基址 ：0x4006_0000

基址	寄存器

+ 地址	+3	+2	+1	+0
0x0000	DMACR[B,H,W]			
	00-00000 -------- -------- --------			
0x0010	DMACA0[B,H,W]			
	00000000 0---0000 00000000 00000000			
0x0014	DMACB0[B,H,W]			
	--000000 00000000 00000000 -------0			
0x0018	DMACSA0[B,H,W]			
	00000000 00000000 00000000 00000000			
0x001C	DMACDA0[B,H,W]			
	00000000 00000000 00000000 00000000			
0x0020	DMACA1[B,H,W]			
	00000000 0---0000 00000000 00000000			
0x0024	DMACB1[B,H,W]			
	--000000 00000000 00000000 -------0			
0x0028	DMACSA1[B,H,W]			
	00000000 00000000 00000000 00000000			
0x002C	DMACDA1[B,H,W]			
	00000000 00000000 00000000 00000000			
0x0030	DMACA2[B,H,W]			
	00000000 0---0000 00000000 00000000			
0x0034	DMACB2[B,H,W]			
	--000000 00000000 00000000 -------0			
0x0038	DMACSA2[B,H,W]			
	00000000 00000000 00000000 00000000			
0x003C	DMACDA2[B,H,W]			
	00000000 00000000 00000000 00000000			
0x0040	DMACA3[B,H,W]			
	00000000 0---0000 00000000 00000000			
0x0044	DMACB3[B,H,W]			
	--000000 00000000 00000000 -------0			
0x0048	DMACSA3[B,H,W]			
	00000000 00000000 00000000 00000000			
0x004C	DMACDA3[B,H,W]			
	00000000 00000000 00000000 00000000			
0x0050	DMACA4[B,H,W]			
	00000000 0---0000 00000000 00000000			
0x0054	DMACB4[B,H,W]			
	--000000 00000000 00000000 -------0			
0x0058	DMACSA4[B,H,W]			
	00000000 00000000 00000000 00000000			
0x005C	DMACDA4[B,H,W]			
	00000000 00000000 00000000 00000000			
0x0060	DMACA5[B,H,W]			
	00000000 00000000 00000000 00000000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x0064	DMACB5[B,H,W]			
	--000000 00000000 00000000 -------0			
0x0068	DMACSA5[B,H,W]			
	00000000 00000000 00000000 00000000			
0x006C	DMACDA5[B,H,W]			
	00000000 00000000 00000000 00000000			
0x0070	DMACA6[B,H,W]			
	00000000 0---0000 00000000 00000000			
0x0074	DMACB6[B,H,W]			
	--000000 00000000 00000000 -------0			
0x0078	DMACSA6[B,H,W]			
	00000000 00000000 00000000 00000000			
0x007C	DMACDA6[B,H,W]			
	00000000 00000000 00000000 00000000			
0x0080	DMACA7[B,H,W]			
	00000000 0---0000 00000000 00000000			
0x0084	DMACB7[B,H,W]			
	--000000 00000000 00000000 -------0			
0x0088	DMACSA7[B,H,W]			
	00000000 00000000 00000000 00000000			
0x008C	DMACDA7[B,H,W]			
	00000000 00000000 00000000 00000000			
0x0090 - 0x00FC	-	-	-	-

1.30 MTB_DWT

MTB_DWT　　基址 ：0xF000_1000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	CMP_ADDR_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x004	CMP_DATA_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x008	CMP_MASK_START[B,H,W] 00000000 00000000 00000000 00000000			
0x00C	-	-	-	-
0x010	CMP_ADDR_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x014	CMP_DATA_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x018	CMP_MASK_STOP[B,H,W] 00000000 00000000 00000000 00000000			
0x01C	-	-	-	-
0x020	-	-	-	FCT[B,H,W]
				00000000
0x024 - 0xFCC	-	-	-	-
0xFD0	PID4[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0xFD4	PID5[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0xFD8	PID6[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0xFDC	PID7[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0xFE0	PID0[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0xFE4	PID1[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0xFE8	PID2[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0xFEC	PID3[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0xFF0	CID0[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0xFF4	CID1[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0xFF8	CID2[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0xFFC	CID3[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			

1.31 快速 GPIO

快速 GPIO　　基址 ：0xF800_0000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	FPDIR0[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x004	-	-	FPDIR1[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x008	-	-	FPDIR2[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x00C	-	-	FPDIR3[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x010	-	-	FPDIR4[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x014	-	-	FPDIR5[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x018	-	-	FPDIR6[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x01C	-	-	FPDIR7[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x020	-	-	FPDIR8[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x024	-	-	FPDIR9[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x028	-	-	FPDIRA[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x02C	-	-	FPDIRB[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x030	-	-	FPDIRC[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x034	-	-	FPDIRD[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x038	-	-	FPDIRE[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x03C	-	-	FPDIRF[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x040	-	-	FPDOR0[B,H,W]	
			00000000 00000000	
0x044	-	-	FPDOR1[B,H,W]	
			00000000 00000000	

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x048	-	-	FPDOR2[B,H,W]	
			00000000 00000000	
0x04C	-	-	FPDOR3[B,H,W]	
			00000000 00000000	
0x050	-	-	FPDOR4[B,H,W]	
			00000000 00000000	
0x054	-	-	FPDOR5[B,H,W]	
			00000000 00000000	
0x058	-	-	FPDOR6[B,H,W]	
			00000000 00000000	
0x05C	-	-	FPDOR7[B,H,W]	
			00000000 00000000	
0x060	-	-	FPDOR8[B,H,W]	
			00000000 00000000	
0x064	-	-	FPDOR9[B,H,W]	
			00000000 00000000	
0x068	-	-	FPDORA[B,H,W]	
			00000000 00000000	
0x06C	-	-	FPDORB[B,H,W]	
			00000000 00000000	
0x070	-	-	FPDORC[B,H,W]	
			00000000 00000000	
0x074	-	-	FPDORD[B,H,W]	
			00000000 00000000	
0x078	-	-	FPDORE[B,H,W]	
			00000000 00000000	
0x07C	-	-	FPDORF[B,H,W]	
			00000000 00000000	
0x080	-	-	-	M_FPDIR0[B,H,W]
				XXXXXXXX
0x084	-	-	-	M_FPDIR1[B,H,W]
				XXXXXXXX
0x088	-	-	-	M_FPDIR2[B,H,W]
				XXXXXXXX
0x08C	-	-	-	M_FPDIR3[B,H,W]
				XXXXXXXX
0x090	-	-	-	M_FPDIR4[B,H,W]
				XXXXXXXX
0x094	-	-	-	M_FPDIR5[B,H,W]
				XXXXXXXX
0x098	-	-	-	M_FPDIR6[B,H,W]
				XXXXXXXX

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x09C	-	-	-	M_FPDIR7[B,H,W]
				XXXXXXXX
0x0A0	-	-	-	M_FPDIR8[B,H,W]
				XXXXXXXX
0x0A4	-	-	-	M_FPDIR9[B,H,W]
				XXXXXXXX
0x0A8	-	-	-	M_FPDIRA[B,H,W]
				XXXXXXXX
0x0AC	-	-	-	M_FPDIRB[B,H,W]
				XXXXXXXX
0x0B0	-	-	-	M_FPDIRC[B,H,W]
				XXXXXXXX
0x0B4	-	-	-	M_FPDIRD[B,H,W]
				XXXXXXXX
0x0B8	-	-	-	M_FPDIRE[B,H,W]
				XXXXXXXX
0x0BC	-	-	-	M_FPDIRF[B,H,W]
				XXXXXXXX
0x0C0	-	-	-	M_FPDOR0[B,H,W]
				00000000
0x0C4	-	-	-	M_FPDOR1[B,H,W]
				00000000
0x0C8	-	-	-	M_FPDOR2[B,H,W]
				00000000
0x0CC	-	-	-	M_FPDOR3[B,H,W]
				00000000
0x0D0	-	-	-	M_FPDOR4[B,H,W]
				00000000
0x0D4	-	-	-	M_FPDOR5[B,H,W]
				00000000
0x0D8	-	-	-	M_FPDOR6[B,H,W]
				00000000
0x0DC	-	-	-	M_FPDOR7[B,H,W]
				00000000
0x0E0	-	-	-	M_FPDOR8[B,H,W]
				00000000
0x0E4	-	-	-	M_FPDOR9[B,H,W]
				00000000
0x0E8	-	-	-	M_FPDORA[B,H,W]
				00000000
0x0EC	-	-	-	M_FPDORB[B,H,W]
				00000000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x0F0	-	-	-	M_FPDORC[B,H,W]
				00000000
0x0F4	-	-	-	M_FPDORD[B,H,W]
				00000000
0x0F8	-	-	-	M_FPDORE[B,H,W]
				00000000
0x0FC	-	-	-	M_FPDORF[B,H,W]
				00000000
0x100 - 0xFFC	-	-	-	-

C. 寄存器映射(TYPE 2-M0+)

本章说明寄存器映射图。

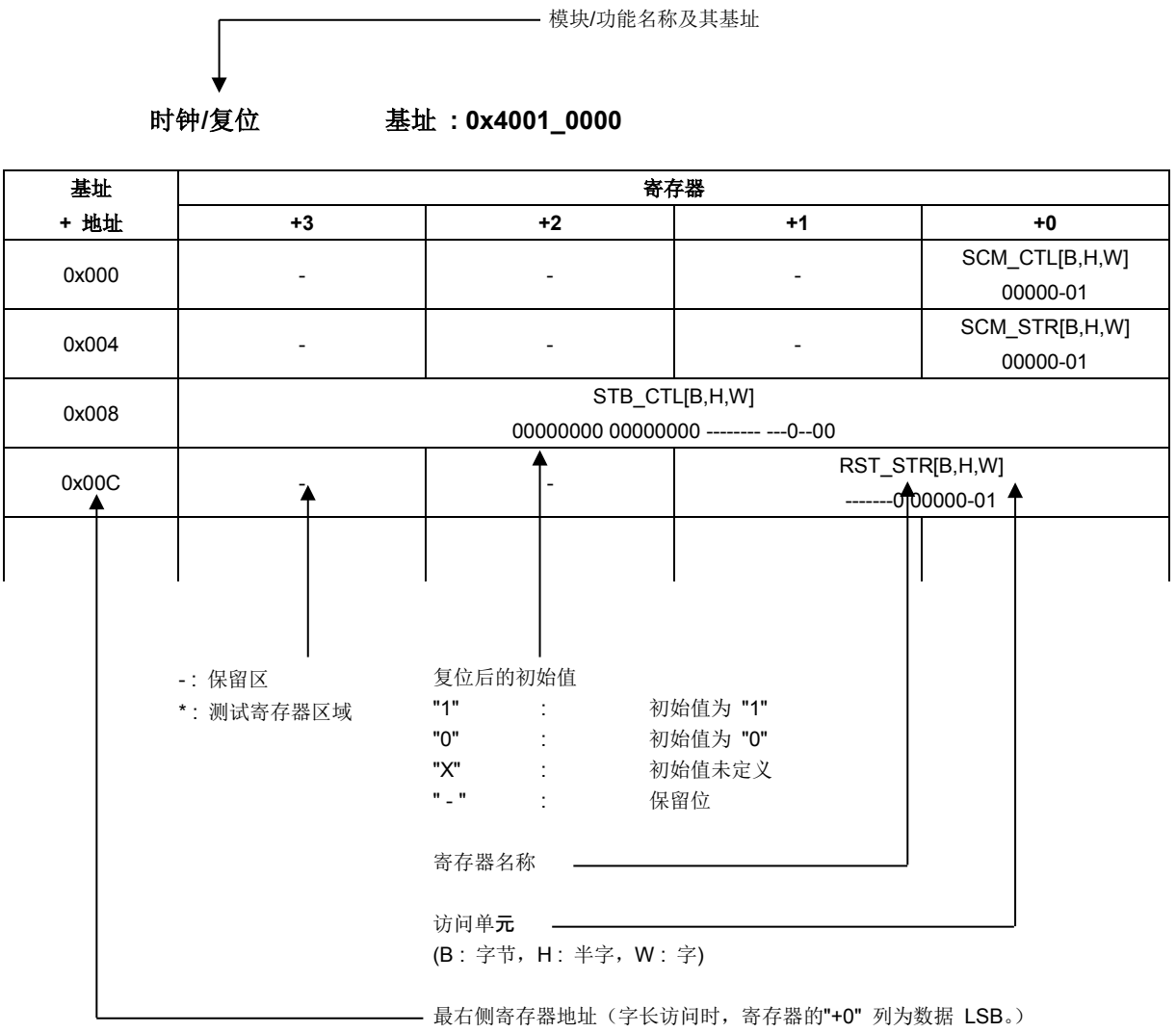
1. 寄存器映射

代码：9AFREGMAP-C01.0

1. 寄存器映射

以下为说明寄存器映射的模块/功能表。

[如何阅读各表]



注意事项：

- 寄存器表用小端表示。
- 执行数据访问时，地址按访问量如下：
- 字访问： ： 地址应为 4 的倍数（最低有效 2 位应为"0x00"）
- 半字访问 ： 地址应为 2 的倍数（最低有效位应为"0x0"）
- 字节访问 ： -
- 不可访问测试寄存器区。

- 不可访问寄存器表中未写入的区域。
- 如果寄存器的访问单元大于寄存器容量，同时也访问保留区，则读取值未定义，写入无效。

1.1 闪存 I/F

闪存 I/F 基址 : 0x4000_0000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	-	-
0x004	FRWTR[B,H,W]			
0x008	FSTR[B,H,W]			
0x00C	FRVRC[B,H,W]			
0x010	FSYNDN[B,H,W]			
0x014 - 0x01C	-	-	-	-
0x020	FICR[B,H,W]			
0x024	FISR[B,H,W]			
0x028	FICLR[B,H,W]			
0x02C - 0x0FC	-	-	-	-
0x100	CRTRMM[B,H,W]			
0x104 - 0x1FC	-	-	-	-

注意事项:

- *有关闪存 I/F 寄存器的详细信息，参见所用产品的《闪存编程手册》。*

1.2 唯一 ID

唯一 ID 基址 : 0x4000_0200

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	UIDR0[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXX----			
0x004	UIDR1[W]			
	-------- -------- ---XXXXX XXXXXXXX			
0x008 – 0xDFC	-	-	-	-

1.3 时钟/复位

时钟/复位 基址 ：0x4001_0000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	-	SCM_CTL[W]
				00000-01
0x004	-	-	-	SCM_STR[W]
				00000-01
0x008	STB_CTL[W]			
	00000000 00000000 -------- ---0-000			
0x00C	-	-	RST_STR[W]	
			-------0 0000--01	
0x010	-	-	-	BSC_PSR[W]
				-----000
0x014	-	-	-	APBC0_PSR[W]
				------00
0x018	-	-	-	APBC1_PSR[W]
				1--0--00
0x01C	-	-	-	-
0x020	-	-	-	SWC_PSR[W]
				X-----00
0x024 - 0x02C	-	-	-	-
0x030	-	-	-	CSW_TMR[W]
				00000000
0x034	-	-	-	PSW_TMR[W]
				---0-000
0x038	-	-	-	PLL_CTL1[W]
				00000000
0x03C	-	-	-	PLL_CTL2[W]
				--000000
0x040	-	-	CSV_CTL[W]	
			-111--00 ------11	
0x044	-	-	-	CSV_STR[W]
				------00
0x048	-	-	FCSWH_CTL[W]	
			11111111 11111111	
0x04C	-	-	FCSWL_CTL[W]	
			00000000 00000000	
0x050	-	-	FCSWD_STR[W]	
			00000000 00000000	
0x054	-	-	-	DBWDT_CTL[W]
				0-0-----

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x058	-	-	-	*
0x05C	-	-	-	-
0x060	-	-	-	INT_ENR[W]
				--0--000
0x064	-	-	-	INT_STR[W]
				--0--000
0x068	-	-	-	INT_CLR[W]
				--0--000
0x06C - 0xFFC	-	-	-	-

1.4 HW WDT

HW WDT 基址 ：0x4001_1000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	WDG_LDR[W]			
	00000000 00000000 11111111 11111111			
0x004	WDG_VLR[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x008	-	-	-	WDG_CTL[W]
				------11
0x00C	-	-	-	WDG_ICL[W]
				XXXXXXXX
0x010	-	-	-	WDG_RIS[R]
				-------0
0x014 - 0xBFC	-	-	-	-
0xC00	WDG_LCK[W]			
	00000000 00000000 00000000 00000001			
0xC04 - 0xFFC	-	-	-	-

1.5 SW_WDT

SW WDT 基址 ：0x4001_2000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	WdogLoad[W]			
	11111111 11111111 11111111 11111111			
0x004	WdogValue[W]			
	11111111 11111111 11111111 11111111			
0x008	-	-	-	WdogControl[W]
				---00000
0x00C	WdogIntClr[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x010	-	-	-	WdogRIS[R]
				-------0
0x014	*			
0x018	-	-	-	WdogSPMC[W]
				-------0
0x01C - 0xBFC	-	-	-	-
0xC00	WdogLock[W]			
	00000000 00000000 00000000 00000000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0xC04 - 0xDFC	-	-	-	-
0xF00	*			
0xF08 - 0xFDF	-	-	-	-
0xFE0 - 0xFFC	*			

1.6 双计时器

双计时器 基址 ：0x4001_5000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	Timer1Load[W]			
	00000000 00000000 00000000 00000000			
0x004	Timer1Value[W]			
	11111111 11111111 11111111 11111111			
0x008	Timer1Control[W]			
	-------- -------- -------- 00100000			
0x00C	Timer1IntClr[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x010	Timer1RIS[W]			
	-------- -------- -------- -------0			
0x014	Timer1MIS[W]			
	-------- -------- -------- -------0			
0x018	Timer1BGLoad[W]			
	00000000 00000000 00000000 00000000			
0x020	Timer2Load[W]			
	00000000 00000000 00000000 00000000			
0x024	Timer2Value[W]			
	11111111 11111111 11111111 11111111			
0x028	Timer2Control[W]			
	-------- -------- -------- 00100000			
0x02C	Timer2IntClr[W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x030	Timer2RIS[W]			
	-------- -------- -------- -------0			
0x034	Timer2MIS[W]			
	-------- -------- -------- -------0			
0x038	Timer2BGLoad[W]			
	00000000 00000000 00000000 00000000			
0x040 - 0xFFC	-	-	-	-

1.7 MFT

MFT 单元 0　　基址 ：0x4002_0000

MFT 单元 1　　基址 ：0x4002_1000

MFT 单元 2　　基址 ：0x4002_2000

基址 + 地址	寄存器			
	+3	+2	+1	+0
0x100	OCCP0[H,W] 00000000 00000000		-	-
0x104	OCCP1[H,W] 00000000 00000000		-	-
0x108	OCCP2[H,W] 00000000 00000000		-	-
0x10C	OCCP3[H,W] 00000000 00000000		-	-
0x110	OCCP4[H,W] 00000000 00000000		-	-
0x114	OCCP5[H,W] 00000000 00000000		-	-
0x118	OCSD10[B,H,W] --000000 00000000		OCSB10[B,H,W] 00000000	OCSA10[B,H,W] 00000000
0x11C	OCSD32[B,H,W] --000000 00000000		OCSB32[B,H,W] 00000000	OCSA32[B,H,W] 00000000
0x120	OCSD54[B,H,W] --000000 00000000		OCSB54[B,H,W] 00000000	OCSA54[B,H,W] 00000000
0x124	-	-	OCSC[B,H,W] --000000	-
0x128	-	-	OCSE0[B,H,W] 00000000 00000000	
0x12C	OCSE1[B,H,W] 00000000 00000000 00000000 00000000			
0x130	-	-	OCSE2[B,H,W] 00000000 00000000	
0x134	OCSE3[B,H,W] 00000000 00000000 00000000 00000000			
0x138	-	-	OCSE4[B,H,W] 00000000 00000000	
0x13C	OCSE5[B,H,W] 00000000 00000000 00000000 00000000			
0x140	TCCP0[H,W] 11111111 11111111		-	-

基址 + 地址	寄存器			
	+3	+2	+1	+0
0x144	TCDT0[H,W] 00000000 00000000		-	-
0x148	TCSC0[H,W] 00000000 00000000		TCSA0[B,H,W] 00000000 01000000	
0x14C	TCCP1[H,W] 11111111 11111111		-	-

基址 + 地址	寄存器			
	+3	+2	+1	+0
0x150	TCDT1[H,W] 00000000 00000000		-	-
0x154	TCSC1[H,W] 00000000 00000000		TCSA1[B,H,W] 00000000 01000000	
0x158	TCCP2[H,W] 11111111 11111111		-	-
0x15C	TCDT2[H,W] 00000000 00000000		-	-
0x160	TCSC2[H,W] 00000000 00000000		TCSA2[B,H,W] 00000000 01000000	
0x164	TCAL[W] 00000000 00000000 11111111 11111111 *1			
	-	-	-	- *2
	*1 MFT unit0 *2 MFT unit1,unit2			
0x168	-	OCFS54[B,H,W] 00000000	OCFS32[B,H,W] 00000000	OCFS10[B,H,W] 00000000
0x16C	-	-	ICFS32[B,H,W] 00000000	ICFS10[B,H,W] 00000000
0x170	-	ACFS54[B,H,W] 00000000	ACFS32[B,H,W] 00000000	ACFS10[B,H,W] 00000000
0x174	ICCP0[H,W] 00000000 00000000		-	-
0x178	ICCP1[H,W] 00000000 00000000		-	-
0x17C	ICCP2[H,W] 00000000 00000000		-	-
0x180	ICCP3[H,W] 00000000 00000000		-	-
0x184	-	-	ICSB10[B,H,W] ------00	ICSA10[B,H,W] 00000000
0x188			ICSB32[B,H,W] ------00	ICSA32[B,H,W] 00000000

基址 + 地址	寄存器			
	+3	+2	+1	+0
0x18C	WFTF10[H,W] 00000000 00000000		-	-
0x190	WFTB10[H,W] 00000000 00000000		WFTA10[H,W] 00000000 00000000	
0x194	WFTF32[H,W] 00000000 00000000		-	-
0x198	WFTB32[H,W] 00000000 00000000		WFTA32[H,W] 00000000 00000000	
0x19C	WFTF54[H,W] 00000000 00000000		-	-

基址 + 地址	寄存器			
	+3	+2	+1	+0
0x1A0	WFTB54[H,W] 00000000 00000000		WFTA54[H,W] 00000000 00000000	
0x1A4	-	-	WFSA10[B,H,W] --000000 000000	
0x1A8	-	-	WFSA32[B,H,W] --000000 000000	
0x1AC	-	-	WFSA54[B,H,W] --000000 000000	
0x1B0	-	-	WFIR[H,W] 00000000 00000000	
0x1B4	-	-	NZCL[H,W] 00000000 00000000	
0x1B8	ACMP0[H,W] 00000000 00000000		-	-
0x1BC	ACMP1[H,W] 00000000 00000000		-	-
0x1C0	ACMP2[H,W] 00000000 00000000		-	-
0x1C4	ACMP3[H,W] 00000000 00000000		-	-
0x1C8	ACMP4[H,W] 00000000 00000000		-	-
0x1CC	ACMP5[H,W] 00000000 00000000		-	-

基址 + 地址	寄存器			
	+3	+2	+1	+0
0x1D0	-	-	ACSA[B,H,W] 00000000 00000000	
0x1D4	-	ACMC0[B,H,W] 00--0000	ACSD0[B,H,W] 00000000	ACSC0[B,H,W] 00000000
0x1D8	-	ACMC1[B,H,W] 00--0000	ACSD1[B,H,W] 00000000	ACSC1[B,H,W] 00000000
0x1DC	-	ACMC2[B,H,W] 00--0000	ACSD2[B,H,W] 00000000	ACSC2[B,H,W] 00000000
0x1E0	-	ACMC3[B,H,W] 00--0000	ACSD3[B,H,W] 00000000	ACSC3[B,H,W] 00000000
0x1E4	-	ACMC4[B,H,W] 00--0000	ACSD4[B,H,W] 00000000	ACSC4[B,H,W] 00000000
0x1E8	-	ACMC5[B,H,W] 00--0000	ACSD5[B,H,W] 00000000	ACSC5[B,H,W] 00000000
0x1EC	-	-	-	TCSD[B,H,W] ------00
0x1F0-0xFFC	-	-	-	-

1.8 PPG

PPG 基址 ：0x4002_4000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	TTCR0[B,H,W] 11110000	-
0x004	-	-	-	*
0x008	-	-	COMP0[B,H,W] 00000000	-
0x00C	-	-	-	COMP2[B,H,W] 00000000
0x010	-	-	COMP4[B,H,W] 00000000	-
0x014	-	-	-	COMP6[B,H,W] 00000000
0x018 - 0x01C	-	-	-	-
0x020	-	-	TTCR1[B,H,W] 11110000	-
0x024	-	-	-	*
0x028	-	-	COMP1[B,H,W] 00000000	-
0x02C	-	-	-	COMP3[B,H,W] 00000000
0x030	-	-	COMP5[B,H,W] 00000000	-
0x034	-	-	-	COMP7[B,H,W] 00000000
0x038 - 0x03C	-	-	-	-
0x040	-	-	TTCR2[B,H,W] 11110000	-
0x044	-	-	-	*
0x048	-	-	COMP8[B,H,W] 00000000	-
0x04C	-	-	-	COMP10[B,H,W] 00000000
0x050	-	-	COMP12[B,H,W] 00000000	-
0x054	-	-	-	COMP14[B,H,W] 00000000
0x058 - 0x0FC	-	-	-	-
0x100	-	-	TRG0[B,H,W] 00000000 00000000	

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x104	-	-	REVC0[B,H,W]	
			00000000 00000000	
0x108 - 0x13C	-	-	-	-
0x140	-	-	TRG1[B,H,W]	
			-------- 00000000	
0x144	-	-	REVC1[B,H,W]	
			-------- 00000000	

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x148 - 0x1FC	-	-	-	-
0x200	-	-	PPGC0[B,H,W]	PPGC1[B,H,W]
			00000000	00000000
0x204	-	-	PPGC2[B,H,W]	PPGC3[B,H,W]
			00000000	00000000
0x208	-	-	PRLH0[B,H,W]	PRLL0[B,H,W]
			XXXXXXXX	XXXXXXXX
0x20C	-	-	PRLH1[B,H,W]	PRLL1[B,H,W]
			XXXXXXXX	XXXXXXXX
0x210	-	-	PRLH2[B,H,W]	PRLL2[B,H,W]
			XXXXXXXX	XXXXXXXX
0x214	-	-	PRLH3[B,H,W]	PRLL3[B,H,W]
			XXXXXXXX	XXXXXXXX
0x218	-	-	-	GATEC0[B,H,W]
				--00---00
0x21C - 0x23C	-	-	-	-
0x240	-	-	PPGC4[B,H,W]	PPGC5[B,H,W]
			00000000	00000000
0x244	-	-	PPGC6[B,H,W]	PPGC7[B,H,W]
			00000000	00000000
0x248	-	-	PRLH4[B,H,W]	PRLL4[B.H.W]
			XXXXXXXX	XXXXXXXX
0x24C	-	-	PRLH5[B,H,W]	PRLL5[B,H,W]
			XXXXXXXX	XXXXXXXX
0x250	-	-	PRLH6[B,H,W]	PRLL6[B,H,W]
			XXXXXXXX	XXXXXXXX
0x254	-	-	PRLH7[B,H,W]	PRLL7[B,H,W]
			XXXXXXXX	XXXXXXXX
0x258	-	-	-	GATEC4[B,H,W]
				--00--00
0x25C - 0x27C	-	-	-	-
0x280	-	-	PPGC8[B,H,W]	PPGC9[B,H,W]
			00000000	00000000
0x284	-	-	PPGC10[B,H,W]	PPGC11[B,H,W]
			00000000	00000000
0x288	-	-	PRLH8[B,H,W]	PRLL8[B,H,W]
			XXXXXXXX	XXXXXXXX
0x28C	-	-	PRLH9[B,H,W]	PRLL9[B,H,W]
			XXXXXXXX	XXXXXXXX
0x290	-	-	PRLH10[B,H,W]	PRLL10[B,H,W]
			XXXXXXXX	XXXXXXXX
0x294	-	-	PRLH11[B,H,W]	PRLL11[B,H,W]
			XXXXXXXX	XXXXXXXX

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x298	-	-	-	GATEC8[B,H,W]
				--00--00
0x29C - 0x2BC	-	-	-	-
0x2C0	-	-	PPGC12[B,H,W]	PPGC13[B,H,W]
			00000000	00000000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x2C4	-	-	PPGC14[B,H,W]	PPGC15[B,H,W]
			00000000	00000000
0x2C8	-	-	PRLH12[B,H,W]	PRLL12[B,H,W]
			XXXXXXXX	XXXXXXXX
0x2CC	-	-	PRLH13[B,H,W]	PRLL13[B,H,W]
			XXXXXXXX	XXXXXXXX
0x2D0	-	-	PRLH14[B,H,W]	PRLL14[B,H,W]
			XXXXXXXX	XXXXXXXX
0x2D4	-	-	PRLH15[B,H,W]	PRLL15[B,H,W]
			XXXXXXXX	XXXXXXXX
0x2D8	-	-	-	GATEC12[B,H,W]
				--00--00
0x2DC - 0x2FC	-	-	-	-
0x300	-	-	PPGC16[B,H,W]	PPGC17[B,H,W]
			00000000	00000000
0x304	-	-	PPGC18[B,H,W]	PPGC19[B,H,W]
			00000000	00000000
0x308	-	-	PRLH16[B,H,W]	PRLL16[B,H,W]
			XXXXXXXX	XXXXXXXX
0x30C	-	-	PRLH17[B,H,W]	PRLL17[B,H,W]
			XXXXXXXX	XXXXXXXX
0x310	-	-	PRLH18[B,H,W]	PRLL18[B,H,W]
			XXXXXXXX	XXXXXXXX
0x314	-	-	PRLH19[B,H,W]	PRLL19[B,H,W]
			XXXXXXXX	XXXXXXXX
0x318	-	-	-	GATEC16[B,H,W]
				--00---00
0x31C - 0x33C	-	-	-	-
0x340	-	-	PPGC20[B,H,W]	PPGC21[B,H,W]
			00000000	00000000
0x344	-	-	PPGC22[B,H,W]	PPGC23[B,H,W]
			00000000	00000000
0x348	-	-	PRLH20[B,H,W]	PRLL20[B,H,W]
			XXXXXXXX	XXXXXXXX
0x34C	-	-	PRLH21[B,H,W]	PRLL21[B,H,W]
			XXXXXXXX	XXXXXXXX
0x350	-	-	PRLH22[B,H,W]	PRLL22[B,H,W]
			XXXXXXXX	XXXXXXXX
0x354	-	-	PRLH23[B,H,W]	PRLL23[B,H,W]
			XXXXXXXX	XXXXXXXX
0x358	-	-	-	GATEC20[B,H,W]
				--00--00
0x35C - 0x37C	-	-	-	-

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x380	-	-	-	IGBTC[B,H,W]
				00000000
0x384 - 0xFFC	-	-	-	-

1.9 基本计时器

基本计时器 ch.0 基址： 0x4002_5000

基本计时器 ch.1 基址： 0x4002_5040

基本计时器 ch.2 基址： 0x4002_5080

基本计时器 ch.3 基址： 0x4002_50C0

基本计时器 ch.4 基址： 0x4002_5200

基本计时器 ch.5 基址： 0x4002_5240

基本计时器 ch.6 基址： 0x4002_5280

基本计时器 ch.7 基址： 0x4002_52C0

基本计时器 ch.8 基址： 0x4002_5400

基本计时器 ch.9 基址： 0x4002_5440

基本计时器 ch.10 基址： 0x4002_5480

基本计时器 ch.11 基址： 0x4002_54C0

基本计时器 ch.12 基址： 0x4002_5600

基本计时器 ch.13 基址： 0x4002_5640

基本计时器 ch.14 基址： 0x4002_5680

基本计时器 ch.15 基址： 0x4002_56C0

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	PCSR/PRLL[H,W]	
			XXXXXXXX XXXXXXXX	
0x004	-	-	PDUT/PRLH/DTBF[H,W]	
			XXXXXXXX XXXXXXXX	
0x008	-	-	TMR[H,W]	
			00000000 00000000	
0x00C	-	-	TMCR[B,H,W]	
			-0000000 00000000	
0x010	-	-	TMCR2[B,H,W]	STC[B,H,W]
			-------0	0000-000
0x014 - 0x03C	-	-	-	-

1.10 基本计时器的 IO 选择器

ch.0-ch.3（基本计时器）的 IO 选择器基址： 0x4002_5100

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	BTSEL0123[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

ch.4-ch.7（基本计时器）的 IO 选择器基址： 0x4002_5300

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	BTSEL4567[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

ch.8-ch.11（基本计时器）的 IO 选择器 基址： 0x4002_5500

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	BTSEL89AB[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

ch.12-ch.15（基本计时器）的 IO 选择器 基址： 0x4002_5700

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	BTSELCDEF[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

基于软件的同时启动（基本计时器）　基址： 0x4002_5F00

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000 - 0x0FB	-	-	-	-
0x0FC	-	-	BTSSSR[B,H,W]	
			XXXXXXXX XXXXXXXX	

1.11 QPRC

QPRC ch.0　　基址： 0x4002_6000

QPRC ch.1　　基址： 0x4002_6040

QPRC ch.2　　基址： 0x4002_6080

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x0000	-	-	QPCR[H,W]	
			00000000 00000000	
0x0004	-	-	QRCR[H,W]	
			00000000 00000000	
0x0008	-	-	QPCCR[H,W]	
			00000000 00000000	
0x000C	-	-	QPRCR[H,W]	
			00000000 00000000	
0x0010	-	-	QMPR[H,W]	
			11111111 11111111	
0x0014	-	-	QICRH[B,H,W]	QICRL[B,H,W]
			--000000	00000000
0x0018	-	-	QCRH[B,H,W]	QCRL[B,H,W]
			00000000	00000000
0x001C	-	-	QECR[B,H,W]	
			-------- -----000	
0x0020 - 0x0038	-	-	-	-
0x003C	QPCRR[B,H,W]		QRCRR[B,H,W]	
	00000000 00000000		00000000 00000000	

1.12 QPRC NF

QPRC ch.0 NF 基址： 0x4002_6100

QPRC ch.1 NF 基址： 0x4002_6110

QPRC ch.2 NF 基址： 0x4002_6120

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x0000	NFCTLA[B,H,W]			
	-------- -------- -------- --00-000			
0x0004	NFCTLB[B,H,W]			
	-------- -------- -------- --00-000			
0x0008	NFCTLC[B,H,W]			
	-------- -------- -------- --00-000			
0x000C	-	-	-	-

1.13 A/DC

12 位 A/DC 单元 0 基址 : 0x4002_7000

12 位 A/DC 单元 1 基址 : 0x4002_7100

12 位 A/DC 单元 2 基址 : 0x4002_7200

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	ADCR[B,H,W]	ADSR[B,H,W]
			000-0000	00---000
0x004	-	-	-	*
0x008	-	-	SCCR[B,H,W]	SFNS[B,H,W]
			1000-000	----0000
0x00C	SCFD[B,H,W]			
	XXXXXXXX XXXX---- ---X--XX ---XXXXX			
0x010	-	-	SCIS3[B,H,W]	SCIS2[B,H,W]
			00000000	00000000
0x014	-	-	SCIS1[B,H,W]	SCIS0[B,H,W]
			00000000	00000000
0x018	-	-	PCCR[B,H,W]	PFNS[B,H,W]
			1000-000	--XX--00
0x01C	PCFD[B,H,W]			
	XXXXXXXX XXXX---- ---X-XXX ---XXXXX			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x020	-	-	-	PCIS[B,H,W]
				00000000
0x024	CMPD[B,H,W]		-	CMPCR[B,H,W]
	00000000 00------			00000000
0x028	-	-	ADSS3[B,H,W]	ADSS2[B,H,W]
			00000000	00000000
0x02C	-	-	ADSS1[B,H,W]	ADSS0[B,H,W]
			00000000	00000000
0x030	-	-	ADST0[B,H,W]	ADST1[B,H,W]
			00010000	00010000
0x034	-	-	-	ADCT[B,H,W]
				00000111
0x038	-	-	SCTSL[B,H,W]	PRTSL[B,H,W]
			----0000	----0000
0x03C	-	-	ADCEN[B,H,W]	
			11111111 ------00	
0x040	*			
0x044	WCMRCIF[B,H,W]			
	-------- -------- -------- -------0			
0x048	WCMRCOT[B,H,W]			
	-------- -------- -------- -------0			
0x04C	-	-	WCMPSR[B,H,W]	WCMRPCR[B,H,W]
			--000000	001000--
0x050	WCMPDH[B,H,W]		WCMPDL[B,H,W]	
	00000000 00------		00000000 00------	
0x054 - 0x0FC	-	-	-	-

1.14 D/AC

10 位 D/AC 基址 ：0x4002_8000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x00	-	DACR0[B,H,W]	DADR0[B,H,W]	
		-------0	------XX XXXXXXXX	
0x04	-	DACR1[B,H,W]	DADR1[B,H,W]	
		-------0	------XX XXXXXXXX	
0x08 - 0xFC	-	-	-	-

1.15 CR Trim

CR Trim 基址 ：0x4002_E000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	-	MCR_PSR[B,H,W]
				------001
0x004	-	-	MCR_FTRM[B,H,W]	
			------01 11101111	
0x008	-	-	-	MCR_TTRM[B,H,W]
				-0111111
0x00C	MCR_RLR[B,H,W]			
	00000000 00000000 00000000 00000001			
0x010 - 0x0FC	-	-	-	-

1.16 EXTI

EXTI 基址 ：0x4003_0000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	ENIR[B,H,W]			
	00000000 00000000 00000000 00000000			
0x004	EIRR[B,H,W]			
	XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x008	EICL[B,H,W]			
	11111111 11111111 11111111 11111111			
0x00C	ELVR[R/W]			
	00000000 00000000 00000000 00000000			
0x010	ELVR1[R/W]			
	00000000 00000000 00000000 00000000			
0x014	-	-	NMIRR[B,H,W]	
			-------- -------0	
0x018	-	-	NMICL[B,H,W]	
			-------- -------1	
0x01C - 0x0FC	-	-	-	-

1.17 INT-Req. 读取

INT-Req. 读取 基址 ：0x4003_1000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-			
	-	-	-	-
0x004	*			
0x008- 0x00B	-	-	-	-
0x00C	-	-	-	IRQCMODE[B,H,W]
				-------0
0x010	EXC02MON[B,H,W]			
	-------- -------- -------- ------00			
0x014	IRQ00MON[B,H,W]			
	-------- -------- -------- -------0			
0x018	IRQ01MON[B,H,W]			
	-------- -------- -------- -------0			
0x01C	IRQ02MON[B,H,W]			
	-------- -------- -------- -------0			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x020	IRQ03MON[B,H,W]			
	-------- -------- -------- ----0000			
0x024	IRQ04MON[B,H,W]			
	-------- -------- -------- 00000000			
0x028	IRQ05MON[B,H,W]			
	-------- -------- 00000000 00000000			
0x02C	IRQ06MON[B,H,W]			
	-------- -------- -------- ------00			
0x030	IRQ07MON[B,H,W]			
	-------- -------- -------- -------0			
0x034	IRQ08MON[B,H,W]			
	-------- -------- -------- ------00			
0x038	IRQ09MON[B,H,W]			
	-------- -------- -------- -------0			
0x03C	IRQ10MON[B,H,W]			
	-------- -------- -------- ------00			
0x040	IRQ11MON[B,H,W]			
	-------- -------- -------- -------0			
0x044	IRQ12MON[B,H,W]			
	-------- -------- -------- ------00			
0x048	IRQ13MON[B,H,W]			
	-------- -------- -------- -------0			
0x04C	IRQ14MON[B,H,W]			
	-------- -------- -------- ------00			
0x050	IRQ15MON[B,H,W]			
	-------- -------- -------- -------0			
0x054	IRQ16MON[B,H,W]			
	-------- -------- -------- ------00			
0x058	IRQ17MON[B,H,W]			
	-------- -------- -------- -------0			
0x05C	IRQ18MON[B,H,W]			
	-------- -------- -------- ------00			
0x060	IRQ19MON[B,H,W]			
	-------- -------- -------- -------0			
0x064	IRQ20MON[B,H,W]			
	-------- -------- -------- ------00			
0x068	IRQ21MON[B,H,W]			
	-------- -------- -------- -------0			
0x06C	IRQ22MON[B,H,W]			
	-------- -------- -------- ------00			
0x070	IRQ23MON[B,H,W]			
	-------- -------- ---0000- -----000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x074	IRQ24MON[B,H,W]			
	-------- -------- -----00- --000000			
0x078	IRQ25MON[B,H,W]			
	-------- -------- -------- ---00000			
0x07C	IRQ26MON[B,H,W]			
	-------- -------- -------- --000000			
0x080	IRQ27MON[B,H,W]			
	-------- -------- -------- --0-----			
0x084	IRQ28MON[B,H,W]			
	-------- -------- -------- --000000			
0x088	IRQ29MON[B,H,W]			
	-------- -------- -------0 00000000			
0x08C	IRQ30MON[B,H,W]			
	-------- -------- -------- --000000			
0x090	IRQ31MON[B,H,W]			
	----0--- -------- 00000000 00000000			
0x094 - 0x20C	-	-	-	-
0x210	RCINTSEL0[B,H,W] 00000000 00000000 00000000 00000000			
0x214	RCINTSEL1[B,H,W] 00000000 00000000 00000000 00000000			
0x218 - 0xFFC	-	-	-	-

1.18 LCDC

LCDC 基址 ：0x4003_2000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x00	-	LCDCC3[B,H,W]	LCDCC2[B,H,W]	LCDCC1[B,H,W]
		0011111-	--010100	-00000--
0x04	LCDC_PSR[B,H,W]			
	-------- -0000000 00000000 00000000			
0x08	LCDC_COMEN[B,H,W]			
	-------- -------- -------- 00000000			
0x0C	LCDC_SEGEN1[B,H,W]			
	00000000 00000000 00000000 00000000			
0x10	LCDC_SEGEN2[B,H,W]			
	-------- -------- -------- 00000000			
0x14	-	-	LCDC_BLINK[B,H,W]	
			00000000 00000000	
0x18	-	-	LCDC_BOOSTER[B,H,W]	
			--001110	----0011
0x1C	LCDRAM03[B,H,W]	LCDRAM02[B,H,W]	LCDRAM01[B,H,W]	LCDRAM00[B,H,W]
	00000000	00000000	00000000	00000000
0x20	LCDRAM07[B,H,W]	LCDRAM06[B,H,W]	LCDRAM05[B,H,W]	LCDRAM04[B,H,W]
	00000000	00000000	00000000	00000000
0x24	LCDRAM11[B,H,W]	LCDRAM10[B,H,W]	LCDRAM09[B,H,W]	LCDRAM08[B,H,W]
	00000000	00000000	00000000	00000000
0x28	LCDRAM15[B,H,W]	LCDRAM14[B,H,W]	LCDRAM13[B,H,W]	LCDRAM12[B,H,W]
	00000000	00000000	00000000	00000000
0x2C	LCDRAM19[B,H,W]	LCDRAM18[B,H,W]	LCDRAM17[B,H,W]	LCDRAM16[B,H,W]
	00000000	00000000	00000000	00000000
0x30	LCDRAM23[B,H,W]	LCDRAM22[B,H,W]	LCDRAM21[B,H,W]	LCDRAM20[B,H,W]
	00000000	00000000	00000000	00000000
0x34	LCDRAM27[B,H,W]	LCDRAM26[B,H,W]	LCDRAM25[B,H,W]	LCDRAM24[B,H,W]
	00000000	00000000	00000000	00000000
0x38	LCDRAM31[B,H,W]	LCDRAM30[B,H,W]	LCDRAM29[B,H,W]	LCDRAM28[B,H,W]
	00000000	00000000	00000000	00000000
0x3C	LCDRAM35[B,H,W]	LCDRAM34[B,H,W]	LCDRAM33[B,H,W]	LCDRAM32[B,H,W]
	00000000	00000000	00000000	00000000
0x40	LCDRAM39[B,H,W]	LCDRAM38[B,H,W]	LCDRAM37[B,H,W]	LCDRAM36[B,H,W]
	00000000	00000000	00000000	00000000
0x44 - 0xFC	-	-	-	-

1.19 GPIO

GPIO 基址 ：0x4003_3000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	PFR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0001 1111			
0x004	PFR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x008	PFR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x00C	PFR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x010	PFR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x014	PFR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x018	PFR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x01C	PFR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x020	PFR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x024	PFR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x028	PFRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x02C	PFRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x030	PFRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x034	PFRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x038	PFRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x03C	PFRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x040 - 0x0FC	-	-	-	-
0x100	PCR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 1010			
0x104	PCR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x108	PCR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x10C	PCR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x110	PCR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x114	PCR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x118	PCR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x11C	PCR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x120	-	-	-	-
0x124	PCR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x128	PCRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x12C	PCRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x130	PCRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x134	PCRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x138	PCRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x13C	PCRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x140 - 0x1FC	-	-	-	-
0x200	DDR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x204	DDR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x208	DDR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x20C	DDR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x210	DDR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x214	DDR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x218	DDR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x21C	DDR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x220	DDR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x224	DDR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x228	DDRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x22C	DDRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x230	DDRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x234	DDRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x238	DDRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x23C	DDRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x240 - 0x2FC	-	-	-	-
0x300	PDIR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x304	PDIR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x308	PDIR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x30C	PDIR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x310	PDIR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x314	PDIR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x318	PDIR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x31C	PDIR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x320	PDIR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x324	PDIR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x328	PDIRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x32C	PDIRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x330	PDIRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x334	PDIRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x338	PDIRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x33C	PDIRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x340 - 0x3FC	-	-	-	-
0x400	PDOR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x404	PDOR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x408	PDOR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x40C	PDOR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x410	PDOR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x414	PDOR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x418	PDOR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x41C	PDOR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x420	PDOR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x424	PDOR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x428	PDORA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x42C	PDORB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x430	PDORC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x434	PDORD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x438	PDORE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x43C	PDORF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x440 - 0x4FC	-	-	-	-
0x500	ADE[B,H,W]			
	---- ---- 1111 1111 1111 1111 1111 1111			
0x504 - 0x57C	-	-	-	-
0x580	SPSR[B,H,W]			
	---- ---- ---- ---- ---- ---- ---- 0101			
0x584 - 0x5FC	-	-	-	-
0x600	EPFR00[B,H,W]			
	---- ---- ---- ---1 ---- ---- 0000 -000			
0x604	EPFR01[B,H,W]			
	0000 0000 0000 0000 --00 0000 0000 0000			
0x608	EPFR02[B,H,W]			
	0000 0000 0000 0000 ---0 0000 0000 0000			
0x60C	EPFR03[B,H,W]			
	0000 0000 0000 0000 ---0 0000 0000 0000			
0x610	EPFR04[B,H,W]			
	--00 0000 --00 00-- --00 0000 -000 00--			
0x614	EPFR05[B,H,W]			
	--00 0000 --00 00-- --00 0000 --00 00--			
0x618	EPFR06[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x61C	EPFR07[B,H,W]			
	---- 0000 0000 0000 0000 0000 0000 ----			
0x620	EPFR08[B,H,W]			
	---- 0000 0000 0000 0000 0000 0000 0000			
0x624	EPFR09[B,H,W]			
	---- ---- ---- ---- 0000 ---- ---- ----			
0x628 - 0x638	-	-	-	-
0x63C	EPFR15[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x640	EPFR16[B,H,W]			
	---- 0000 0000 0000 0000 0000 0000 0000			
0x644	-			
	-	-	-	-
0x648	EPFR18[B,H,W]			
	--00 0000 0000 0000 0000 0000 0000 0000			
0x64C - 0x658	-	-	-	-
0x65C	EPFR23[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x660 - 0x680	-	-	-	-
0x684	EPFR33[B,H,W]			
	---- 0000 0000 0000 ---- 0000 0000 0000			
0x688	EPFR34[B,H,W]			
	---- ---- ---- ---- ---- ---- 0000 ----			
0x68C - 0x690	-	-	-	-
0x694	EPFR37[B,H,W]			
	---- 0000 0000 0000 ---- ---- ---- ----			
0x698	EPFR38[B,H,W]			
	---- ---- ---- ---- ---- 0000 0000 0000			
0x69C - 0x6FC	-	-	-	-
0x700	PZR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x704	PZR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x708	PZR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x70C	PZR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x710	PZR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x714	PZR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x718	PZR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x71C	PZR7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x720	PZR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x724	PZR9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x728	PZRA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x72C	PZRB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x730	PZRC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x734	PZRD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x738	PZRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x73C	PZRF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x740	LVDIE[B,H,W]			
	---- ---- ---- ---- ---- ---- ---- ---0			
0x744 - 0x7FC	-	-	-	-
0x800	*			
0x804	*			
0x808 - 0x8FC	-	-	-	-
0x900	FPOER0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x904	FPOER1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x908	FPOER2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x90C	FPOER3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x910	FPOER4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x914	FPOER5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x918	FPOER6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x91C	FPOER7[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x920	FPOER8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x924	FPOER9[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x928	FPOERA[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x92C	FPOERB[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x930	FPOERC[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x934	FPOERD[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x938	FPOERE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x93C	FPOERF[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x940 - 0xFFC	-	-	-	-

1.20 HDMI-CEC

HDMI-CEC/远控接收器 ch.0 基址 : 0x4003_4000

HDMI-CEC/远控接收器 ch.1 基址 : 0x4003_4100

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x00	-	-	-	TXCTRL[B,H,W]
				000000-0
0x04	-	-	-	TXDATA[B,H,W]
				00000000
0x08	-	-	-	TXSTS[B,H,W]
				0-00---0
0x0C	-	-	-	SFREE[B,H,W]
				----0000
0x10 - 0x3F	-	-	-	-
0x40	-	-	RCCR[B,H,W]	RCST[B,H,W]
			0---0000	00000000
0x44	-	-	RCSHW[B,H,W]	RCDAHW[B,H,W]
			00000000	00000000
0x48	-	-	RCDBHW[B,H,W]	-
			00000000	
0x4C	-	-	RCADR1[B,H,W]	RCADR2[B,H,W]
			---00000	---00000
0x50	-	-	RCDTHH[B,H,W]	RCDTHL[B,H,W]
			00000000	00000000
0x54	-	-	RCDTLH[B,H,W]	RCDTLL[B,H,W]
			00000000	00000000
0x58	-	-	RCCKD[H,W]	
			---00000 00000000	
0x5C	-	-	RCRC[B,H,W]	RCRHW[B,H,W]
			---0---0	00000000
0x60	-	-	RCLE[B,H,W]	-
			00000-00	
0x64	-	-	RCLELW[B,H,W]	RCLESW[B,H,W]
			00000000	00000000
0x68 - 0xFC	-	-	-	-

1.21 LVD

LVD 基址 ：0x4003_5000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	LVD_CTL [B,H,W]	
			100000-- 000011--	
0x004	-	-	LVD_STR [B,H,W]	
			0------1 0------1	
0x008	-	-	LVD_CLR [B,H,W]	
			1------- 1-------	
0x00C	LVD_RLR [W]			
	---------- 00000000 00000001			
0x010	-	-	LVD_STR2 [B,H,W]	
			0------- 01------	
0x014	-	-	LVD_CTL2 [B,H,W]	
			-------0 000011--	
0x018	-	-	-	LVD2_CTL [B,H,W]
				000011--
0x01C	-	-	LVD2_CTL2 [B,H,W]	
			0------0 000011--	
0x020 - 0x0FC	-	-	-	-

1.22 DS 模式

DS 模式　基址 ：0x4003_5100

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	-	REG_CTL [B,H,W]
				---- -10-
0x004	-	-	-	RCK_CTL [B,H,W]
				---- --01
0x008	-	-	-	REG_CTL2 [B,H,W]
				---- -011
0x00C	-	-	-	-
0x010	-	-	-	MOSC_CTL [B,H,W]
				---- -10-
0x014 - 0x0FC	-	-	-	-
0x100	-	-	-	CAL_CTL [B,H,W]
				---- 0000
0x104	-	-	-	CAL_SET [B,H,W]
				---1 0001
0x108	CAL_KEY [W]			
	00000000 00000000 00000000 00000001			
0x10C - 0x1FC	-	-	-	-
0x200 - 0x6FC	-	-	-	-
0x700	-	-	-	PMD_CTL [B,H,W]
				---- ---0
0x704	-	-	-	WRFSR [B,H,W]
				---- --00
0x708	-	-	WIFSR [B,H,W]	
			00000000 00000000	
0x70C	-	-	WIER [B,H,W]	
			00000000 00000-00	
0x710	-	-	WILVR [B,H,W]	
			-----000 00000000	
0x714	-	-	-	DSRAMR [B,H,W]
				---- --00
0x718 - 0x71C	-	-	-	-
0x720	-	-	-	STBFLASHPDX [B,H,W]
				---- ---0
0x724	RST_MSK [W]			
	00000000 00000000 -------- -------0			
0x728 - 0x7FC	-	-	-	-

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x800	BUR04 [B,H,W]	BUR03 [B,H,W]	BUR02 [B,H,W]	BUR01 [B,H,W]
	00000000	00000000	00000000	00000000
0x804	BUR08 [B,H,W]	BUR07 [B,H,W]	BUR06 [B,H,W]	BUR05 [B,H,W]
	00000000	00000000	00000000	00000000
0x808	BUR012 [B,H,W]	BUR11 [B,H,W]	BUR10 [B,H,W]	BUR09 [B,H,W]
	00000000	00000000	00000000	00000000
0x80C	BUR16 [B,H,W]	BUR15 [B,H,W]	BUR14 [B,H,W]	BUR13 [B,H,W]
	00000000	00000000	00000000	00000000
0x810 - 0xEFC	-	-	-	-

1.23 USB 时钟

USB 时钟 基址 ：0x4003_6000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	-	UCCR [B,H,W]
				-000 0000
0x004	-	-	-	UPCR [B,H,W]
				---- --00
0x008	-	-	-	UPCR2 [B,H,W]
				---- -000
0x00C	-	-	-	UPCR3 [B,H,W]
				---0 0000
0x010	-	-	-	UPCR4 [B,H,W]
				-0111011
0x014	-	-	-	UP_STR [B,H,W]
				---- ---0
0x018	-	-	-	UPINT_ENR [B,H,W]
				---- ---0
0x01C	-	-	-	UPINT_CLR [B,H,W]
				---- ---0
0x020	-	-	-	UPINT_STR [B,H,W]
				---- ---0
0x024	-	-	-	UPCR5 [B,H,W]
				---- 0001
0x028	-	-	-	UPCR6 [B,H,W]
				---- 0010
0x02C	-	-	-	UP_CR7 [B,H,W]
				---- ---0
0x030	-	-	-	USBEN0 [B,H,W]
				---- -100

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x034	-	-	-	USBEN1 [B,H,W]
				---- -100
0x038 - 0xFFC	-	-	-	-

1.24 MFS

MFS ch.0 基址 : 0x4003_8000

MFS ch.1 基址 : 0x4003_8100

MFS ch.2 基址 : 0x4003_8200

MFS ch.3 基址 : 0x4003_8300

MFS ch.4 基址 : 0x4003_8400

MFS ch.5 基址 : 0x4003_8500

MFS ch.6 基址 : 0x4003_8600

MFS ch.7 基址 : 0x4003_8700

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	SCR/IBCR[B,H,W]	SMR[B,H,W]
			0--00000	000-00-0
0x004	-	-	SSR[B,H,W]	ESCR/IBSR[B,H,W]
			0-000011	00000000
0x008	RDR/TDR[H,W]			
	00000000 00000000 00000000 00000000			
0x00C	-	-	BGR1[B,H,W]	BGR0[B,H,W]
			00000000	00000000
0x010	-	-	ISMK[B,H,W]	ISBA[B,H,W]
			--------	--------
0x014	-	-	FCR1[B,H,W]	FCR0[B,H,W]
			---00100	-0000000
0x018	-	-	FBYTE2[B,H,W]	FBYTE1[B,H,W]
			00000000	00000000
0x01C	-	-	SCSTR1/ EIBCR[B,H,W]	SCSTR0[B,H,W]
			00000000	00000000
0x020	-	-	SCSTR3[B,H,W]	SCSTR2[B,H,W]
			00000000	00000000
0x024	-	-	SACSR[B,H,W]	
			--000--0 00-00000	
0x028	-	-	STMR[B,H,W]	
			00000000 00000000	
0x02C	-	-	STMCR[B,H,W]	
			00000000 00000000	

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x030	-	-	SCSCR[B,H,W]	
			00000000 00100000	
0x034	-	-	SCSFR1[B,H,W] 10000000	SCSFR0[B,H,W] 10000000
0x038	-	-	-	SCSFR2[B,H,W] 10000000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x03C	-	-	TBYTE1[B,H,W]	TBYTE0[B,H,W]
			00000000	00000000
0x040	-	-	TBYTE3[B,H,W]	TBYTE2[B,H,W]
			00000000	00000000
0x044	-	-	FTICR2[B,H,W]	FTICR1[B,H,W]
			00000000	00000000
0x048 - 0x0FC	-	-	-	-

注意事项：

- *RDR/TDR 寄存器的高 16 位可通过 MFS-I²S 模式下的字操作进行访问。*

1.25 CRC

CRC 基址 ：0x4003_9000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	-	CRCCR[B,H,W]
				-0000000
0x004	CRCINIT[B,H,W]			
	11111111 11111111 11111111 11111111			
0x008	CRCIN[B,H,W]			
	00000000 00000000 00000000 00000000			
0x00C	CRCR[B,H,W]			
	11111111 11111111 11111111 11111111			

1.26 计时计数器

计时计数器 基址 ：0x4003_A000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	WCCR[B,H,W]	WCRL[B,H,W]	WCRD[B,H,W]
		00--0000	--000000	--000000
0x004 - 0x00C	-	-	-	-
0x010	-	-	CLK_SEL[B,H,W]	
			-----000 ------00	
0x014	-	-	-	CLK_EN[B,H,W]
				------00
0x018 - 0xFFC	-	-	-	-

1.27 RTC

RTC 基址 ：0x4003_B000

基址+ 地址	寄存器			
	+3	+2	+1	+0
0x100	-	-	-	WTCR10[B,H,W] 00000000
0x104	-	-	-	WTCR11[B,H,W] ---00000
0x108	-	-	-	WTCR12[B,H,W] 00000000
0x10C	-	-	-	WTCR13[B,H,W] 00000000
0x110	-	-	-	WTCR20[B,H,W] --000000
0x114	-	-	-	WTCR21[B,H,W] -----000
0x118	-	-	-	*
0x11C	-	-	-	WTSR[B,H,W] -0000000
0x120	-	-	-	WTMIR[B,H,W] -0000000
0x124	-	-	-	WTHR[B,H,W] --000000
0x128	-	-	-	WTDR[B,H,W] --000000
0x12C	-	-	-	WTDW[B,H,W] -----000
0x130	-	-	-	WTMOR[B,H,W] ---00000
0x134	-	-	-	WTYR[B,H,W] 00000000
0x138	-	-	-	ALMIR[B,H,W] -0000000
0x13C	-	-	-	ALHR[B,H,W] --000000
0x140	-	-	-	ALDR[B,H,W] --000000
0x144	-	-	-	ALMOR[B,H,W] ---00000
0x148	-	-	-	ALYR[B,H,W] 00000000

基址+ 地址	寄存器			
	+3	+2	+1	+0
0x14C	-	-	-	WTTR0[B,H,W] 00000000
0x150	-	-	-	WTTR1[B,H,W] 00000000
0x154	-	-	-	WTTR2[B,H,W] ------00
0x158	-	-	-	WTCAL0[B,H,W] 00000000
0x15C	-	-	-	WTCAL1[B,H,W] ------00
0x160	-	-	-	WTCALEN[B,H,W] -------0
0x164	-	-	-	WTDIV[B,H,W] ----0000
0x168	-	-	-	WTDIVEN[B,H,W] ------00
0x16C	-	-	-	WTCALPRD[B,H,W] --010011
0x170	-	-	-	WTCOSEL[B,H,W] -------0
0x174	-	-	-	VB_CLKDIV[B,H,W] 00000111
0x178	-	-	-	WTOSCCNT[B,H,W] ------01
0x17C	-	-	-	CCS[B,H,W] 11001110
0x180	-	-	-	CCB[B,H,W] 11001110
0x184	-	-	-	*
0x188	-	-	-	BOOST[B,H,W] ------11
0x18C	-	-	-	EWKUP[B,H,W] -------0
0x190	-	-	-	VDET[B,H,W] 00------
0x194	-	-	-	*
0x198	-	-	-	HIBRST[B,H,W] -------0
0x19C	-	-	-	VBPFR[B,H,W] --011100
0x1A0	-	-	-	VBPCR[B,H,W] ----0000

基址+ 地址	寄存器			
	+3	+2	+1	+0
0x1A4	-	-	-	VBDDR[B,H,W] ----XXXX
0x1A8	-	-	-	VBDIR[B,H,W] ----0000
0x1AC	-	-	-	VBDOR[B,H,W] ----1111
0x0B0	-	-	-	VBPZR[B,H,W] ------11
0x1B4-1FF	-	-	-	-
0x200	BREG03[B,H,W] 00000000	BREG02[B,H,W] 00000000	BREG01[B,H,W] 00000000	BREG00[B,H,W] 00000000
0x204	BREG07[B,H,W] 00000000	BREG06[B,H,W] 00000000	BREG05[B,H,W] 00000000	BREG04[B,H,W] 00000000
0x208	BREG0B[B,H,W] 00000000	BREG0A[B,H,W] 00000000	BREG09[B,H,W] 00000000	BREG08[B,H,W] 00000000
0x20C	BREG0F[B,H,W] 00000000	BREG0E[B,H,W] 00000000	BREG0D[B,H,W] 00000000	BREG0C[B,H,W] 00000000
0x210	BREG13[B,H,W] 00000000	BREG12[B,H,W] 00000000	BREG11[B,H,W] 00000000	BREG10[B,H,W] 00000000
0x214	BREG17[B,H,W] 00000000	BREG16[B,H,W] 00000000	BREG15[B,H,W] 00000000	BREG14[B,H,W] 00000000
0x218	BREG1B[B,H,W] 00000000	BREG1A[B,H,W] 00000000	BREG19[B,H,W] 00000000	BREG18[B,H,W] 00000000
0x21C	BREG1F[B,H,W] 00000000	BREG1E[B,H,W] 00000000	BREG1D[B,H,W] 00000000	BREG1C[B,H,W] 00000000
0x220	BREG23[B,H,W] 00000000	BREG22[B,H,W] 00000000	BREG21[B,H,W] 00000000	BREG20[B,H,W] 00000000
0x224	BREG27[B,H,W] 00000000	BREG26[B,H,W] 00000000	BREG25[B,H,W] 00000000	BREG24[B,H,W] 00000000
0x228	BREG2B[B,H,W] 00000000	BREG2A[B,H,W] 00000000	BREG29[B,H,W] 00000000	BREG28[B,H,W] 00000000
0x22C	BREG2F[B,H,W] 00000000	BREG2E[B,H,W] 00000000	BREG2D[B,H,W] 00000000	BREG2C[B,H,W] 00000000
0x230	BREG33[B,H,W] 00000000	BREG32[B,H,W] 00000000	BREG31[B,H,W] 00000000	BREG30[B,H,W] 00000000
0x234	BREG37[B,H,W] 00000000	BREG36[B,H,W] 00000000	BREG35[B,H,W] 00000000	BREG34[B,H,W] 00000000
0x238	BREG3B[B,H,W] 00000000	BREG3A[B,H,W] 00000000	BREG39[B,H,W] 00000000	BREG38[B,H,W] 00000000
0x23C	BREG3F[B,H,W] 00000000	BREG3E[B,H,W] 00000000	BREG3D[B,H,W] 00000000	BREG3C[B,H,W] 00000000
0x240	BREG43[B,H,W] 00000000	BREG42[B,H,W] 00000000	BREG41[B,H,W] 00000000	BREG40[B,H,W] 00000000

基址+ 地址	寄存器			
	+3	+2	+1	+0
0x244	BREG47[B,H,W] 00000000	BREG46[B,H,W] 00000000	BREG45[B,H,W] 00000000	BREG44[B,H,W] 00000000
0x248	BREG4B[B,H,W] 00000000	BREG4A[B,H,W] 00000000	BREG49[B,H,W] 00000000	BREG48[B,H,W] 00000000
0x24C	BREG4F[B,H,W] 00000000	BREG4E[B,H,W] 00000000	BREG4D[B,H,W] 00000000	BREG4C[B,H,W] 00000000
0x250	BREG53[B,H,W] 00000000	BREG52[B,H,W] 00000000	BREG51[B,H,W] 00000000	BREG50[B,H,W] 00000000
0x254	BREG57[B,H,W] 00000000	BREG56[B,H,W] 00000000	BREG55[B,H,W] 00000000	BREG54[B,H,W] 00000000
0x258	BREG5B[B,H,W] 00000000	BREG5A[B,H,W] 00000000	BREG59[B,H,W] 00000000	BREG58[B,H,W] 00000000
0x25C	BREG5F[B,H,W] 00000000	BREG5E[B,H,W] 00000000	BREG5D[B,H,W] 00000000	BREG5C[B,H,W] 00000000
0x260	BREG63[B,H,W] 00000000	BREG62[B,H,W] 00000000	BREG61[B,H,W] 00000000	BREG60[B,H,W] 00000000
0x264	BREG67[B,H,W] 00000000	BREG66[B,H,W] 00000000	BREG65[B,H,W] 00000000	BREG64[B,H,W] 00000000
0x268	BREG6B[B,H,W] 00000000	BREG6A[B,H,W] 00000000	BREG69[B,H,W] 00000000	BREG68[B,H,W] 00000000
0x26C	BREG6F[B,H,W] 00000000	BREG6E[B,H,W] 00000000	BREG6D[B,H,W] 00000000	BREG6C[B,H,W] 00000000
0x270	BREG73[B,H,W] 00000000	BREG72[B,H,W] 00000000	BREG71[B,H,W] 00000000	BREG70[B,H,W] 00000000
0x274	BREG77[B,H,W] 00000000	BREG76[B,H,W] 00000000	BREG75[B,H,W] 00000000	BREG74[B,H,W] 00000000
0x278	BREG7B[B,H,W] 00000000	BREG7A[B,H,W] 00000000	BREG79[B,H,W] 00000000	BREG78[B,H,W] 00000000
0x27C	BREG7F[B,H,W] 00000000	BREG7E[B,H,W] 00000000	BREG7D[B,H,W] 00000000	BREG7C[B,H,W] 00000000
0x280-0xFFC	-	-	-	-

1.28 低速 CR 预分频器

低速 CR 预分频器　基址 ：0x4003_C000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	-	LCR_PRSLD[B,H,W]
				--000000
0x000 - 0x0FC	-	-	-	-

1.29 外设时钟选通

外设时钟选通　　基址 ：0x4003_C100

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	CKEN0[B,H,W]			
	---1--11 ----1111 11111111 11111111			
0x004	MRST0[B,H,W]			
	------00 ----0000 00000000 00000000			
0x008 - 0x00C	-	-	-	-
0x010	CKEN1[B,H,W]			
	-------- ----1111 ----1111 ----1111			
0x014	MRST1[B,H,W]			
	-------- ----0000 ----0000 ----0000			
0x018 - 0x01C	-	-	-	-
0x020	CKEN2[B,H,W]			
	-------- ---1-1-1 1111-1-- --**--00 Products with CAN : *="1" Products without CAN : *="0"			
0x024	MRST2[B,H,W]			
	-------- ---0-0-0 0000-0-- --00--00			
0x028 - 0x0FC	-	-	-	-

1.30 智能卡 I/F

智能卡 I/F ch.0　　基址 ：0x4003_C900

智能卡 I/F ch.1　　基址 ：0x4003_C980

基址 + 地址	寄存器			
	+3	+2	+1	+0
0x00	-	-	GLOBALCONTROL1[H,W] -0001000 00000000	
0x04	-	-	STATUS[H,W] --000000 00000001	
0x08	-	-	PORTCONTROL[H,W] 0000--00 00-0-0-0	
0x0C	-	-	DATA[H,W] -------0 00000000	
0x10	-	-	CARDCLOCK [H,W] 00000000 00101000	
0x14	-	-	BAUDRATE[H,W] 0000001 01110100	
0x18	-	-	GUARDTIMER[H,W] -------- 00000000	
0x1C	-	-	IDLETIMER[H,W] 00000000 00000000	
0x20	-	-	GLOBALCONTROL2[H,W] -------- ----1-00	
0x24	-	-	DATA_FIFO[H,W] -------0 00000000	
0x28	-	-	FIFO_LEVEL_READ[H,W] 00000000 00000000	
0x2C	-	-	FIFO_LEVEL_WRITE[H,W] 00000000 00000000	
0x30	-	-	FIFO_MODE[H,W] 00000000 ----0000	
0x34	-	-	FIFO_CLEAR_MSB_WRITE[H,W] -------- -------0	
0x38	-	-	FIFO_CLEAR_MSB_READ[H,W] -------- -------0	
0x3C	-	-	-	-
0x40	-	-	IRQ_STATUS[H,W] -------- 00000000	
0x44-0x7C	-	-	-	-

1.31 MFSI2S

MFSI2S ch.5　　基址 ：0x4003_CA00

MFSI2S ch.6　　基址 ：0x4003_CA80

基址	寄存器			
+Address	+3	+2	+1	+0
0x00	-		CNTLREG [H,W] -------- -------- -----000 00000001	
0x04	-		I2SCLK [H,W] -------- -------- 000----- 00000000	
0x08	-		I2SST [B] ------00	I2SRST[B] 00000000

1.32 高容错性

高容错性　基址 ：0x4003_E000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	RTR_CTL3 [B,H,W]	RTR_CTL2 [B,H,W]	RTR_CTL1 [B,H,W]	RTR_CTL0 [B,H,W]
	000- 000-	000- ----	---- ----	1111 1111
0x004	RTR_RTS3 [B,H,W]	RTR_RTS2 [B,H,W]	RTR_RTS1 [B,H,W]	RTR_RTS0 [B,H,W]
	1111 1111	1111 1111	1111 1111	1111 1111
0x008	RTR_TGS3 [B,H,W]	RTR_TGS2 [B,H,W]	RTR_TGS1 [B,H,W]	RTR_TGS0 [B,H,W]
	1111 1111	1111 1111	1111 1111	1111 1111
0x00C	RTR_STR3 [B,H,W]	RTR_STR2 [B,H,W]	RTR_STR1 [B,H,W]	RTR_STR0 [B,H,W]
	00-- ----	---- ----	00-- ----	---- ----
0x010	RTR_RLR [W]			
	00000000 00000000 00000000 00000000			
0x014	RTR_CT23 [B,H,W]	RTR_CT22 [B,H,W]	RTR_CT21 [B,H,W]	RTR_CT20 [B,H,W]
	0000 0000	0000 0000	0000 0000	---0 ---0
0x018	RTR_REV [B,H,W]			
	00000000 00010101 00000001 00000000			
0x01C - 0xFFC	-	-	-	-

1.33 USB

USB ch.0 基址 ：0x4004_0000

USB ch.1 基址 ：0x4005_0000

基址+ 地址	寄存器			
	+3	+2	+1	+0
0x2100	-	-	HCNT1[B,H,W] -----001	HCNT0[B,H,W] 00000000
0x2104	-	-	HERR[B,H,W] 00000011	HIRQ[B,H,W] 0-000000
0x2108	-	-	HFCOMP[B,H,W] 00000000	HSTATE[B,H,W] --010010
0x210C	-	-	HRTIMER(1/0)[B,H,W] 0000000000000000	
0x2110	-	-	HADR[B,H,W] -0000000	HRTIMER(2)[B,H,W] ------00
0x2114	-	-	HEOF(1/0)[B,H,W] --00000000000000	
0x2118	-	-	HFRAME(1/0)[B,H,W] -----00000000000	
0x211C	-	-	-	HTOKEN[B,H,W] 00000000
0x2120	-	-	UDCC[B,H,W] -------- 10100-00	
0x2124	-	-	EP0C[H,W] ------0- -1000000	
0x2128	-	-	EP1C[H,W] 01100001 00000000	
0x212C	-	-	EP2C[H,W] 0110000- -1000000	
0x2130	-	-	EP3C[H,W] 0110000- -1000000	
0x2134	-	-	EP4C[H,W] 0110000- -1000000	
0x2138	-	-	EP5C[H,W] 0110000- -1000000	
0x213C	-	-	TMSP[H,W] -----000 00000000	
0x2140	-	-	UDCIE[B,H,W] --000000	UDCS[B,H,W] --000000
0x2144	-	-	EP0IS[H,W] 10---1-- --------	

基址+ 地址	寄存器			
	+3	+2	+1	+0
0x2148	-	-	EP0OS[H,W] 100--00- -XXXXXXX	
0x214C	-	-	EP1S[H,W] 100-000X XXXXXXXX	
0x2150	-	-	EP2S[H,W] 100-000- -XXXXXXX	
0x2154	-	-	EP3S[H,W] 100-000- -XXXXXXX	
0x2158	-	-	EP4S[H,W] 100-000- -XXXXXXX	
0x215C	-	-	EP5S[H,W] 100-000- -XXXXXXX	
0x2160	-	-	EP0DTH[B,H,W] XXXXXXXX	EP0DTL[B,H,W] XXXXXXXX
0x2164	-	-	EP1DTH[B,H,W] XXXXXXXX	EP1DTL[B,H,W] XXXXXXXX
0x2168	-	-	EP2DTH[B,H,W] XXXXXXXX	EP2DTL[B,H,W] XXXXXXXX
0x216C	-	-	EP3DTH[B,H,W] XXXXXXXX	EP3DTL[B,H,W] XXXXXXXX
0x2170	-	-	EP4DTH[B,H,W] XXXXXXXX	EP4DTL[B,H,W] XXXXXXXX
0x2174	-	-	EP5DTH[B,H,W] XXXXXXXX	EP5DTL[B,H,W] XXXXXXXX
0x2178 - 0x217C	-	-	-	-

1.34 DSTC

DSTC 基址 ：0x4006_1000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x0000	DESTP[B,H,W] 00000000 00000000 00000000 00000000			
0x0004	HWDESP[B,H,W] 00XXXXXX XXXXXX00 00000000 00000000			
0x0008	SWTR[H] 00000000 00000000		CFG[B] 01000000	CMD[B] 00000001
0x000C	MONERS[B,H,W] 00XXXXXX XXXXXX00 XXXXXXXX XXX00000			
0x0010	DREQENB[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x0014	DREQENB[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x0018-0x002C	-			
0x0030	HWINT[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x0034	HWINT[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x0038-0x004C	-			
0x0050	HWINTCLR[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x0054	HWINTCLR[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x0058-0x006C	-			
0x070	DQMSK[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x074	DQMSK[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x078-0x08C	-			
0x090	DQMSKCLR[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x094	DQMSKCLR[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x098-0x0FFC	-			

1.35 MTB_DWT

MTB_DWT　　基址 ：0xF000_1000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	CMP_ADDR_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x004	CMP_DATA_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x008	CMP_MASK_START[B,H,W] 00000000 00000000 00000000 00000000			
0x00C	-	-	-	-
0x010	CMP_ADDR_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x014	CMP_DATA_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x018	CMP_MASK_STOP[B,H,W] 00000000 00000000 00000000 00000000			
0x01C	-	-	-	-
0x020	-	-	-	FCT[B,H,W]
				00000000
0x024 - 0xFCC	-	-	-	-
0xFD0	PID4[B,H,W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFD4	PID5[B,H,W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFD8	PID6[B,H,W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFDC	PID7[B,H,W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFE0	PID0[B,H,W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFE4	PID1[B,H,W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFE8	PID2[B,H,W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFEC	PID3[B,H,W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFF0	CID0[B,H,W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFF4	CID1[B,H,W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0xFF8	CID2[B,H,W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0xFFC	CID3[B,H,W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			

1.36 快速 GPIO

快速 GPIO 基址 ：0xF800_0000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	FPDIR0[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x004	-	-	FPDIR1[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x008	-	-	FPDIR2[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x00C	-	-	FPDIR3[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x010	-	-	FPDIR4[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x014	-	-	FPDIR5[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x018	-	-	FPDIR6[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x01C	-	-	FPDIR7[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x020	-	-	FPDIR8[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x024	-	-	FPDIR9[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x028	-	-	FPDIRA[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x02C	-	-	FPDIRB[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x030	-	-	FPDIRC[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x034	-	-	FPDIRD[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x038	-	-	FPDIRE[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x03C	-	-	FPDIRF[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x040	-	-	FPDOR0[B,H,W]	
			00000000 00000000	
0x044	-	-	FPDOR1[B,H,W]	
			00000000 00000000	
0x048	-	-	FPDOR2[B,H,W]	
			00000000 00000000	

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x04C	-	-	FPDOR3[B,H,W]	
			00000000 00000000	
0x050	-	-	FPDOR4[B,H,W]	
			00000000 00000000	
0x054	-	-	FPDOR5[B,H,W]	
			00000000 00000000	

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x058	-	-	FPDOR6[B,H,W]	
			00000000 00000000	
0x05C	-	-	FPDOR7[B,H,W]	
			00000000 00000000	
0x060	-	-	FPDOR8[B,H,W]	
			00000000 00000000	
0x064	-	-	FPDOR9[B,H,W]	
			00000000 00000000	
0x068	-	-	FPDORA[B,H,W]	
			00000000 00000000	
0x06C	-	-	FPDORB[B,H,W]	
			00000000 00000000	
0x070	-	-	FPDORC[B,H,W]	
			00000000 00000000	
0x074	-	-	FPDORD[B,H,W]	
			00000000 00000000	
0x078	-	-	FPDORE[B,H,W]	
			00000000 00000000	
0x07C	-	-	FPDORF[B,H,W]	
			00000000 00000000	
0x080	-	-	-	M_FPDIR0[B,H,W]
				XXXXXXXX
0x084	-	-	-	M_FPDIR1[B,H,W]
				XXXXXXXX
0x088	-	-	-	M_FPDIR2[B,H,W]
				XXXXXXXX
0x08C	-	-	-	M_FPDIR3[B,H,W]
				XXXXXXXX
0x090	-	-	-	M_FPDIR4[B,H,W]
				XXXXXXXX
0x094	-	-	-	M_FPDIR5[B,H,W]
				XXXXXXXX
0x098	-	-	-	M_FPDIR6[B,H,W]
				XXXXXXXX
0x09C	-	-	-	M_FPDIR7[B,H,W]
				XXXXXXXX
0x0A0	-	-	-	M_FPDIR8[B,H,W]
				XXXXXXXX
0x0A4	-	-	-	M_FPDIR9[B,H,W]
				XXXXXXXX
0x0A8	-	-	-	M_FPDIRA[B,H,W]
				XXXXXXXX

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x0AC	-	-	-	M_FPDIRB[B,H,W]
				XXXXXXXX
0x0B0	-	-	-	M_FPDIRC[B,H,W]
				XXXXXXXX
0x0B4	-	-	-	M_FPDIRD[B,H,W]
				XXXXXXXX

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x0B8	-	-	-	M_FPDIRE[B,H,W]
				XXXXXXXX
0x0BC	-	-	-	M_FPDIRF[B,H,W]
				XXXXXXXX
0x0C0	-	-	-	M_FPDOR0[B,H,W]
				00000000
0x0C4	-	-	-	M_FPDOR1[B,H,W]
				00000000
0x0C8	-	-	-	M_FPDOR2[B,H,W]
				00000000
0x0CC	-	-	-	M_FPDOR3[B,H,W]
				00000000
0x0D0	-	-	-	M_FPDOR4[B,H,W]
				00000000
0x0D4	-	-	-	M_FPDOR5[B,H,W]
				00000000
0x0D8	-	-	-	M_FPDOR6[B,H,W]
				00000000
0x0DC	-	-	-	M_FPDOR7[B,H,W]
				00000000
0x0E0	-	-	-	M_FPDOR8[B,H,W]
				00000000
0x0E4	-	-	-	M_FPDOR9[B,H,W]
				00000000
0x0E8	-	-	-	M_FPDORA[B,H,W]
				00000000
0x0EC	-	-	-	M_FPDORB[B,H,W]
				00000000
0x0F0	-	-	-	M_FPDORC[B,H,W]
				00000000
0x0F4	-	-	-	M_FPDORD[B,H,W]
				00000000
0x0F8	-	-	-	M_FPDORE[B,H,W]
				00000000
0x0FC	-	-	-	M_FPDORF[B,H,W]
				00000000
0x100 - 0xFFC	-	-	-	-

D. 寄存器映射(TYPE 3-M0+)

本章说明寄存器映射图。

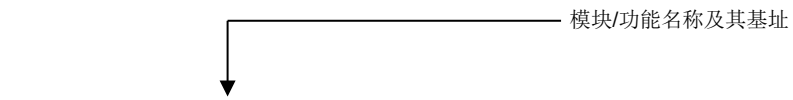
1. 寄存器映射

代码：9AFREGMAP-C03.0

1. 寄存器映射

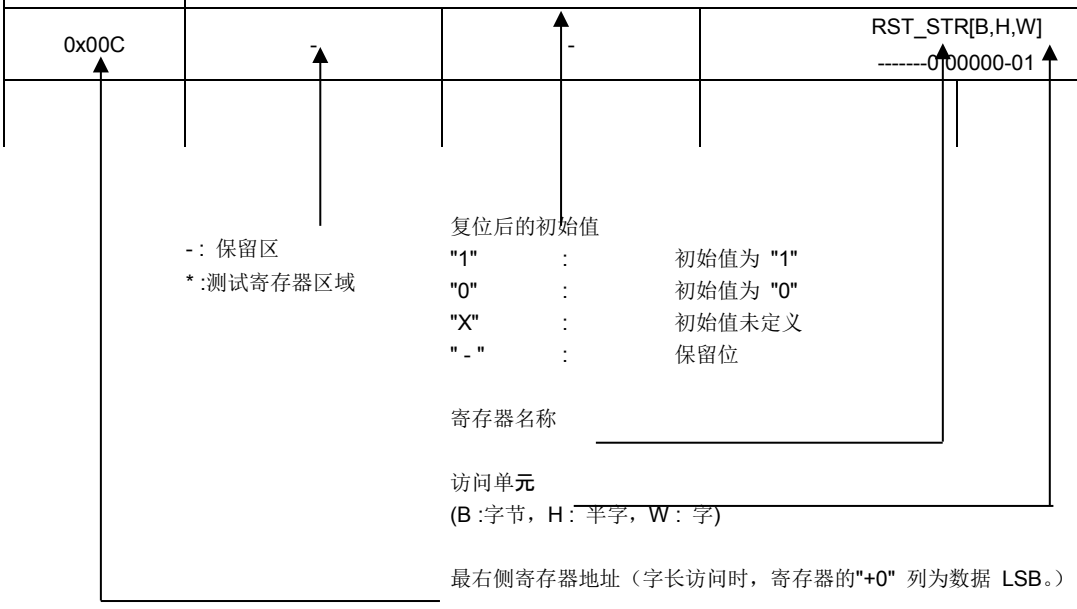
以下为说明寄存器映射的模块/功能表。

[如何阅读各表]



时钟/复位 基址 ：0x4001_0000

基址 + 地址	寄存器			
	+3	+2	+1	+0
0x000	-	-	-	SCM_CTL[B,H,W] 00000-01
0x004	-	-	-	SCM_STR[B,H,W] 00000-01
0x008	STB_CTL[B,H,W] 00000000 00000000 -------- ---0--00			
0x00C	-	-	RST_STR[B,H,W] -------0 00000-01	



注意事项:

- 寄存器表用小端表示。
- 执行数据访问时，地址按访问量如下：
- 字访问 ： 地址应为 4 的倍数（最低有效 2 位应为"0x00"）
- 半字访问 ： 地址应为 2 的倍数（最低有效位应为"0x0"）
- 字节访问 ： -
- 不可访问测试寄存器区。

- 不可访问寄存器表中未写入的区域。
- 如果寄存器的访问单元大于寄存器容量，同时也访问保留区，则读取值未定义，写入无效。

1.1 闪存 I/F

闪存 I/F 基址 : 0x4000_0000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	-	-
0x004	FRWTR[B,H,W]			
	-------- -------- -------- -----011			
0x008	FSTR[B,H,W]			
	-------- -------- -------- --00000X			
0x00C	-	-	-	-
0x010	FSYNDN[B,H,W]			
	-------- -------- -------- ----0001			
0x014 - 0x01C	-	-	-	-
0x020	FICR[B,H,W]			
	-------- -------- -------- ------00			
0x024	FISR[B,H,W]			
	-------- -------- -------- ------00			
0x028	FICLR[B,H,W]			
	-------- -------- -------- ------00			
0x02C - 0x0FC	-	-	-	-
0x100	CRTRMM[B,H,W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x104 - 0x1FC	-	-	-	-

注意事项:

- *有关闪存 I/F 寄存器的详细信息，参见所用产品的《闪存编程手册》。*

1.2 唯一 ID

唯一 ID 基址 : 0x4000_0200

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	UIDR0[W]			
	XXXXXXXX XXXXXXXXXXXXXXXX XXXX----			
0x004	UIDR1[W]			
	-------- -------- ---XXXXX XXXXXXXX			
0x008 – 0xDFC	-	-	-	-

1.3 时钟/复位

时钟/复位 基址 ：0x4001_0000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	-	SCM_CTL[W]
				00000-01
0x004	-	-	-	SCM_STR[W]
				00000-01
0x008	STB_CTL[W]			
	00000000 00000000 -------- ---0-000			
0x00C	-	-	RST_STR[W]	
			-------0 00000-01	
0x010	-	-	-	BSC_PSR[W]
				-----000
0x014	-	-	-	APBC0_PSR[W]
				------00
0x018	-	-	-	APBC1_PSR[W]
				1--0--00
0x01C	-	-	-	-
0x020	-	-	-	SWC_PSR[W]
				X-----00
0x024 - 0x02C	-	-	-	-
0x030	-	-	-	CSW_TMR[W]
				00000000
0x034	-	-	-	PSW_TMR[W]
				---0-000
0x038	-	-	-	PLL_CTL1[W]
				00000000
0x03C	-	-	-	PLL_CTL2[W]
				--000000
0x040	-	-	CSV_CTL[W]	
			-111--00 ------11	
0x044	-	-	-	CSV_STR[W]
				------00
0x048	-	-	FCSWH_CTL[W]	
			11111111 11111111	
0x04C	-	-	FCSWL_CTL[W]	
			00000000 00000000	
0x050	-	-	FCSWD_STR[W]	
			00000000 00000000	
0x054	-	-	-	DBWDT_CTL[W]
				0-0-----

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x058	-	-	-	*
0x05C	-	-	-	-
0x060	-	-	-	INT_ENR[W]
				--0--000
0x064	-	-	-	INT_STR[W]
				--0--000
0x068	-	-	-	INT_CLR[W]
				--0--000
0x06C - 0xFFC	-	-	-	-

1.4 HW WDT

HW WDT 基址 : 0x4001_1000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	WDG_LDR[W]			
	00000000 00000000 11111111 11111111			
0x004	WDG_VLR[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x008	-	-	-	WDG_CTL[W]
				------11
0x00C	-	-	-	WDG_ICL[W]
				XXXXXXXX
0x010	-	-	-	WDG_RIS[W]
				-------0
0x014 - 0xBFC	-	-	-	-
0xC00	WDG_LCK[W]			
	00000000 00000000 00000000 00000001			
0xC04 - 0xFFC	-	-	-	-

1.5 SW WDT

SW WDT 基址 :0x4001_2000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	WdogLoad[W]			
	11111111 11111111 11111111 11111111			
0x004	WdogValue[W]			
	11111111 11111111 11111111 11111111			
0x008	-	-	-	WdogControl[W]
				---00000
0x00C	WdogIntClr[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x010	-	-	-	WdogRIS[W]
				-------0
0x014	*			
0x018	-	-	-	WdogSPMC[W]
				-------0
0x01C - 0xBFC	-	-	-	-
0xC00	WdogLock[W]			
	00000000 00000000 00000000 00000000			

0xC04 - 0xDFC	-	-	-	-
0xF00	*			
0xF08 - 0xFDF	-	-	-	-
0xFE0 - 0xFFC	*			

1.6 双计时器

双计时器　基址 ：**0x4001_5000**

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	Timer1Load[W]			
	00000000 00000000 00000000 00000000			
0x004	Timer1Value[W]			
	11111111 11111111 11111111 11111111			
0x008	Timer1Control[W]			
	-------- -------- -------- 00100000			
0x00C	Timer1IntClr[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x010	Timer1RIS[W]			
	-------- -------- -------- -------0			
0x014	Timer1MIS[W]			
	-------- -------- -------- -------0			
0x018	Timer1BGLoad[W]			
	00000000 00000000 00000000 00000000			
0x020	Timer2Load[W]			
	00000000 00000000 00000000 00000000			
0x024	Timer2Value[W]			
	11111111 11111111 11111111 11111111			
0x028	Timer2Control[W]			
	-------- -------- -------- 00100000			
0x02C	Timer2IntClr[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x030	Timer2RIS[W]			
	-------- -------- -------- -------0			
0x034	Timer2MIS[W]			
	-------- -------- -------- -------0			
0x038	Timer2BGLoad[W]			
	00000000 00000000 00000000 00000000			
0x040 - 0xFFC	-	-	-	-

1.7 基本计时器

基本计时器 ch.0 基址： 0x4002_5000

基本计时器 ch.1 基址： 0x4002_5040

基本计时器 ch.2 基址： 0x4002_5080

基本计时器 ch.3 基址： 0x4002_50C0

基本计时器 ch.4 基址： 0x4002_5200

基本计时器 ch.5 基址： 0x4002_5240

基本计时器 ch.6 基址： 0x4002_5280

基本计时器 ch.7 基址： 0x4002_52C0

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	PCSR/PRLL[H,W]	
			XXXXXXXX XXXXXXXX	
0x004	-	-	PDUT/PRLH/DTBF[H,W]	
			XXXXXXXX XXXXXXXX	
0x008	-	-	TMR[H,W]	
			00000000 00000000	
0x00C	-	-	TMCR[B,H,W]	
			-0000000 00000000	
0x010	-	-	TMCR2[B,H,W]	STC[B,H,W]
			0-------0	0000-000
0x014 - 0x03C	-	-	-	-

1.8 基本计时器的 IO 选择器

ch.0-ch.3（基本计时器）的 IO 选择器基址： 0x4002_5100

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	BTSEL0123[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

ch.4-ch.7（基本计时器）的 IO 选择器基址： 0x4002_5300

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	BTSEL4567[B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

基于软件的同时启动（基本计时器） Base Address : 0x4002_5F00

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000 - 0x0FB	-	-	-	-
0x0FC	-	-	BTSSSR [B,H,W] --------XXXXXXXX	

1.9 A/DC

12 位 A/DC 单元 0 基址 ：0x4002_7000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	ADCR[B,H,W]	ADSR[B,H,W]
			000-0000	00---000
0x004	-	-	-	*
0x008	-	-	SCCR[B,H,W]	SFNS[B,H,W]
			1000-000	----0000
0x00C	SCFD[B,H,W]			
	XXXXXXXX XXXX---- ---X--XX ---XXXXX			
0x010	-	-	SCIS3[B,H,W]	SCIS2[B,H,W]
			00000000	00000000
0x014	-	-	SCIS1[B,H,W]	SCIS0[B,H,W]
			00000000	00000000
0x018	-	-	PCCR[B,H,W]	PFNS[B,H,W]
			10000000	--XX--00
0x01C	PCFD[B,H,W]			
	XXXXXXXX XXXX---- ---X-XXX ---XXXXX			
0x020	-	-	-	PCIS[B,H,W]
				00000000
0x024	CMPD[B,H,W]		-	CMPCR[B,H,W]
	00000000 00------			00000000
0x028	-	-	ADSS3[B,H,W]	ADSS2[B,H,W]
			00000000	00000000
0x02C	-	-	ADSS1[B,H,W]	ADSS0[B,H,W]
			00000000	00000000
0x030	-	-	ADST0[B,H,W]	ADST1[B,H,W]
			00010000	00010000
0x034	-	-	-	ADCT[B,H,W]
				00000111
0x038	-	-	SCTSL[B,H,W]	PRTSL[B,H,W]
			----0000	----0000
0x03C	-	-	ADCEN[B,H,W]	
			11111111 ------00	
0x040	*			
0x044	WCMRCIF[B,H,W]			
	-------- -------- -------- -------0			
0x048	WCMRCOT[B,H,W]			
	-------- -------- -------- -------0			
0x04C	-	-	WCMPSR[B,H,W]	WCMRPCR[B,H,W]
			00000000	001000--

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x050	WCMPDH[B,H,W]		WCMPDL[B,H,W]	
	00000000 00------		00000000 00------	
0x054 - 0x0FC	-	-	-	-

1.10 CR 调节

CR 调节 基址 : 0x4002_E000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	-	MCR_PSR[B,H,W]
				-----001
0x004	-	-	MCR_FTRM[B,H,W]	
			------10 00000110	
0x008	-	-	-	MCR_TTRM[B,H,W]
				-1111111
0x00C	MCR_RLR[B,H,W]			
	00000000 00000000 00000000 00000001			
0x010 - 0x0FC	-	-	-	-

1.11 EXTI

EXTI 基址 : 0x4003_0000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	ENIR[B,H,W]			
	00000000 00000000 00000000 00000000			
0x004	EIRR[B,H,W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x008	EICL[B,H,W]			
	11111111 11111111 11111111 11111111			
0x00C	ELVR[R/W]			
	00000000 00000000 00000000 00000000			
0x010	ELVR1[R/W]			
	00000000 00000000 00000000 00000000			
0x014	-	-	NMIRR[B,H,W]	
			-------- -------0	

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x018	-	-	NMICL[B,H,W]	
			-------- -------1	
0x01C	ELVR2[B,H,W]			
	00000000 00000000 00000000 00000000			
0x020	-	-	-	NMIENR[B,H,W]
	-	-	-	-------0
0x024- 0x0FC	-	-	-	-

1.12 INT-Req. 读取

INT-Req. 读取　　基址 ：0x4003_1000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000 – 0x004	-	-	-	-
0x008	VIR_OFFSET[B,H,W]			
	00000000 00000000 00000000 00000000			
0x010	-	-	-	ODDPKS[B,H,W]
				---00000
0x014– 0x1FC	-	-	-	-
0x200	EXC02MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x204	IRQ00MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x208	IRQ01MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x20C	IRQ02MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x210	IRQ03MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x214	IRQ04MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x218	IRQ05MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x21C	IRQ06MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x220	IRQ07MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x224	IRQ08MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x228	IRQ09MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x22C	IRQ10MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x230	IRQ11MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x234	IRQ12MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x238	IRQ13MON[B,H,W]			
	00000000 00000000 00000000 00000000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x23C	IRQ14MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x240	IRQ15MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x244	IRQ16MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x248	IRQ17MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x24C	IRQ18MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x250	IRQ19MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x254	IRQ20MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x258	IRQ21MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x25C	IRQ22MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x260	IRQ23MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x264	IRQ24MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x268	IRQ25MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x26C	IRQ26MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x270	IRQ27MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x274	IRQ28MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x278	IRQ29MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x27C	IRQ30MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x280	IRQ31MON[B,H,W]			
	00000000 00000000 00000000 00000000			
0x284- 0xFFC	-	-	-	-

1.13 GPIO

GPIO 基址 ：0x4003_3000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	PFR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 1010			
0x004	PFR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x008	PFR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x00C	PFR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x010	PFR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x014	PFR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x018	PFR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x01C	-	-	-	-
0x020	PFR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x024– 0x034	-	-	-	-
0x038	PFRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x03C- 0x0FC	-	-	-	-
0x100	PCR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 1010			
0x104	PCR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x108	PCR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x10C	PCR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x110	PCR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x114	PCR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x118	PCR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x11C – 0x134	-	-	-	-
0x138	PCRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x13C – 0x1FC	-	-	-	-

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x200	DDR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x204	DDR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x208	DDR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x20C	DDR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x210	DDR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x214	DDR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x218	DDR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x21C	-	-	-	-
0x220	DDR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x224 – 0x234	-	-	-	-
0x238	DDRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x23C - 0x2FC	-	-	-	-
0x300	PDIR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x304	PDIR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x308	PDIR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x30C	PDIR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x310	PDIR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x314	PDIR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x318	PDIR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x31C	-	-	-	-
0x320	PDIR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x324– 0x334	-	-	-	-
0x338	PDIRE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x33C - 0x3FC	-	-	-	-

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x400	PDOR0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x404	PDOR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x408	PDOR2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x40C	PDOR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x410	PDOR4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x414	PDOR5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x418	PDOR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x41C	-	-	-	-
0x420	PDOR8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x424– 0x434	-	-	-	-
0x438	PDORE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x43C- 0x4FC	-	-	-	-
0x500	ADE[B,H,W]			
	1111 1111 1111 1111 1111 1111 1111 1111			
0x504 - 0x57C	-	-	-	-
0x580	SPSR[B,H,W]			
	---- ---- ---- ---- ---- ---- ---00101			
0x584 - 0x5FC	-	-	-	-
0x600	EPFR00[B,H,W]			
	---- --00 ---- --01 ---- --0- ---- --00			
0x604 - 0x60C	-	-	-	-
0x610	EPFR04[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x614	EPFR05[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x618	EPFR06[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x61C	EPFR07[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x620	EPFR08[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x624	EPFR09[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x628 - 0x654	-	-	-	-

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x658	EPFR22[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x65C	EPFR23[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x660 - 0x678	-	-	-	-
0x67C	EPFR31[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x680	-	-	-	-
0x684	EPFR33[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x688- 0x690	-	-	-	-
0x694	EPFR37[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x698	EPFR38[B,H,W]			
	0000 0000 0000 0000 0000 0000 0000 0000			
0x69C - 0x6FC	-	-	-	-
0x700	-	-	-	-
0x704	PZR1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x708	-	-	-	-
0x70C	PZR3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x710 – 0x714	-	-	-	-
0x718	PZR6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x71C – 0x7FC	-	-	-	-
0x800– 0x8FC	-	-	-	-
0x900	FPOER0[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x904	FPOER1[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x908	FPOER2[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x90C	FPOER3[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x910	FPOER4[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x914	FPOER5[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x918	FPOER6[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x91C	-	-	-	-

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x920	FPOER8[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x924 – 0x934	-	-	-	-
0x938	FPOERE[B,H,W]			
	---- ---- ---- ---- 0000 0000 0000 0000			
0x93C- 0xFFC	-	-	-	-

1.14 HDMI-CEC

HDMI-CEC/远控接收器 ch.0 基址 : 0x4003_4000

HDMI-CEC/远控接收器 ch.1 基址 : 0x4003_4100

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x00	-	-	-	TXCTRL[B,H,W]
				000000-0
0x04	-	-	-	TXDATA[B,H,W]
				00000000
0x08	-	-	-	TXSTS[B,H,W]
				0-00---0
0x0C	-	-	-	SFREE[B,H,W]
				----0000
0x10 - 0x3F	-	-	-	-
0x40	-	-	RCCR[B,H,W]	RCST[B,H,W]
			0---0000	00000000
0x44	-	-	RCSHW[B,H,W]	RCDAHW[B,H,W]
			00000000	00000000
0x48	-	-	RCDBHW[B,H,W]	-
			00000000	
0x4C	-	-	RCADR1[B,H,W]	RCADR2[B,H,W]
			---00000	---00000
0x50	-	-	RCDTHH[B,H,W]	RCDTHL[B,H,W]
			00000000	00000000
0x54	-	-	RCDTLH[B,H,W]	RCDTLL[B,H,W]
			00000000	00000000
0x58	-	-	RCCKD[H,W]	
			---00000 00000000	
0x5C	-	-	RCRC[B,H,W]	RCRHW[B,H,W]
			---0---0	00000000
0x60	-	-	RCLE[B,H,W]	-
			00000-00	
0x64	-	-	RCLELW[B,H,W]	RCLESW[B,H,W]
			00000000	00000000
0x68 - 0xFC	-	-	-	-

1.15 LVD

LVD 基址 ：0x4003_5000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	LVD_CTL [B,H,W]	
			10000000 00001100	
0x004	-	-	LVD_STR [B,H,W]	
			00000000 0000000-	
0x008	-	-	LVD_CLR [B,H,W]	
			00000000 10000000	
0x00C	LVD_RLR [W]			
	0000000000000000 00000000 00000001			
0x010	-	-	LVD_STR2 [B,H,W]	
			0000000001000000	
0x014 - 0x0FC	-	-	-	-

1.16 DS 模式

DS 模式　基址 ：0x4003_5100

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	-	REG_CTL [B,H,W]
				---- -10-
0x004	-	-	-	RCK_CTL [B,H,W]
				---- --01
0x008 - 0x00C	-	-	-	-
0x010	-	-	-	MOSC_CTL [B,H,W]
				---- -10-
0x014 - 0x0FC	-	-	-	-
0x100	-	-	-	CAL_CTL [B,H,W]
				---- 0000
0x104	-	-	-	-
0x108	CAL_KEY [W]			
	00000000 00000000 00000000 00000001			
0x10C - 0x6FC	-	-	-	-
0x700	-	-	-	PMD_CTL [B,H,W]
				---- ---0
0x704	-	-	-	WRFSR [B,H,W]
				---- --00
0x708	-	-	WIFSR [B,H,W]	
			00000000 00000000	
0x70C	-	-	WIER [B,H,W]	
			00000000 00000-00	
0x710	-	-	WILVR [B,H,W]	
			-----000 00000000	
0x714	-	-	-	DSRAMR [B,H,W]
				---- --00
0x718 - 0x7FC	-	-	-	-
0x800	BUR04 [B,H,W]	BUR03 [B,H,W]	BUR02 [B,H,W]	BUR01 [B,H,W]
	00000000	00000000	00000000	00000000
0x804	BUR08 [B,H,W]	BUR07 [B,H,W]	BUR06 [B,H,W]	BUR05 [B,H,W]
	00000000	00000000	00000000	00000000
0x808	BUR012 [B,H,W]	BUR11 [B,H,W]	BUR10 [B,H,W]	BUR09 [B,H,W]
	00000000	00000000	00000000	00000000
0x80C	BUR16 [B,H,W]	BUR15 [B,H,W]	BUR14 [B,H,W]	BUR13 [B,H,W]
	00000000	00000000	00000000	00000000
0x810 – 0x8FC	-	-	-	-
0x900	WIOLC_CTL [B,H,W]			
	-------0 -------1 -------0 -------0			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x904	-	-	-	SUBOSC_CTL[B,H,W]
				------01
0x908	-	-	-	CEC_CTL [B,H,W]
				----0000
0x90C	-	-	-	DEBUG_SW_CTL[B,H,W]
				-------1
0x910 - 0xEFC	-	-	-	-

1.17 USB 时钟

USB 时钟 基址 ：0x4003_6000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	-	UCCR [B,H,W]
				-----000
0x004 – 0x024	-	-	-	-
0x028	-	-	-	UPCR6[B,H,W]
				----0010
0x02C				
0x030	-	-	-	USBEN0[B,H,W]
				-------0
0x038 – 0x0FC	-	-	-	-

1.18 I2CSLAVE

I2CSLAVE ch.6　　基址 ：0x4003_7980

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x00	IBSCR[B,H,W]		IBSSR[B,H,W]	
	------00 0-000000		-----001 00000000	
0x04	-	IBSDSTUPR[B,H,W]	IBSMSKR[B,H,W]	IBSADR[B,H,W]
	-	11111111	01111111	00000000
0x08	-	-	-	IBSTDR[B,H,W]
	-	-	-	11111111
0x0C	-	-	-	IBSRDR[B,H,W]
	-	-	-	11111111
0x10	-	-	IBSSCR[B,H,W]	
	-	-	-----0-- -----00-	
0x14	-	-	IBSSSR[B,H,W]	
	-	-	-------0 --------	
0x18 - 0x3F	-	-	-	-

1.19 MFS

MFS ch.0 基址 ：0x4003_8000

MFS ch.1 基址 ：0x4003_8100

MFS ch.3 基址 ：0x4003_8300

MFS ch.4 基址 ：0x4003_8400

MFS ch.6 基址 :0x4003_8600

MFS ch.7 基址 ：0x4003_8700

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	SCR/IBCR[B,H,W]	SMR[B,H,W]
			0--00000	000-00-0
0x004	-	-	SSR[B,H,W]	ESCR/IBSR[B,H,W]
			0-000011	00000000
0x008	RDR/TDR[H,W]			
	00000000 0000000000000000 00000000			
0x00C	-	-	BGR1[B,H,W]	BGR0[B,H,W]
			00000000	00000000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x010	-	-	ISMK[B,H,W]	ISBA[B,H,W]
			--------	--------
0x014	-	-	FCR1[B,H,W]	FCR0[B,H,W]
			---00100	-0000000
0x018	-	-	FBYTE2[B,H,W]	FBYTE1[B,H,W]
			00000000	00000000
0x01C	-	-	SCSTR1/ EIBCR[B,H,W]	SCSTR0[B,H,W]
			00000000	00000000
0x020	-	-	SCSTR3[B,H,W]	SCSTR2[B,H,W]
			00000000	00000000
0x024	-	-	SACSR[B,H,W]	
			--000--0 00-00000	
0x028	-	-	STMR[B,H,W]	
			00000000 00000000	
0x02C	-	-	STMCR[B,H,W]	
			00000000 00000000	
0x030	-	-	SCSCR[B,H,W]	
			00000000 00100000	
0x034	-	-	SCSFR1[B,H,W] 10000000	SCSFR0[B,H,W] 10000000
0x038	-	-	-	SCSFR2[B,H,W] 10000000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x03C	-	-	TBYTE1[B,H,W]	TBYTE0[B,H,W]
			00000000	00000000
0x040	-	-	TBYTE3[B,H,W]	TBYTE2[B,H,W]
			00000000	00000000
0x044 - 0x0FC	-	-	-	-

注意事项：

- *RDR/TDR 寄存器的高 16 位可通过 I^2S 模式下的字操作进行访问。*

1.20 CRC

CRC 基址 ：0x4003_9000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	-	CRCCR[B,H,W]
				-0000000
0x004	CRCINIT[B,H,W]			
	11111111 11111111 11111111 11111111			
0x008	CRCIN[B,H,W]			
	00000000 00000000 00000000 00000000			
0x00C	CRCR[B,H,W]			
	11111111 11111111 11111111 11111111			

1.21 计时计数器

计时计数器 基址 :0x4003_A000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	WCCR[B,H,W]	WCRL[B,H,W]	WCRD[B,H,W]
		00--0000	--000000	--000000
0x004 - 0x00C	-	-	-	-
0x010	-	-	CLK_SEL[B,H,W]	
			-----000 ------00	
0x014	-	-	-	CLK_EN[B,H,W]
				------00
0x018 - 0xFFC	-	-	-	-

1.22 RTC

RTC 基址 ：0x4003_B000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	WTCR1[B,H,W]			
	00000000 00000000 ---00000 -00000-0			
0x004	WTCR2[B,H,W]			
	-------- -------- -----000 -------0			
0x008	WTBR[B,H,W]			
	-------- 00000000 00000000 00000000			
0x00C	WTDR[B,H,W]	WTHR[B,H,W]	WTMIR[B,H,W]	WTSR[B,H,W]
	--000000	--000000	-0000000	-0000000
0x010	-	WTYR[B,H,W]	WTMOR[B,H,W]	WTDW[B,H,W]
		00000000	---00000	-----000
0x014	ALDR[B,H,W]	ALHR[B,H,W]	ALMIR[B,H,W]	-
	--000000	--000000	-0000000	
0x018	-	ALYR[B,H,W]	ALMOR[B,H,W]	-
		00000000	---00000	
0x01C	WTTR[B,H,W]			
	-------- ------00 00000000 00000000			
0x020	-	-	WTCLKM[B,H,W]	WTCLKS[B,H,W]
			------00	-------0
0x024	-	WTCALEN[B,H,W]	WTCAL[B,H,W]	
		-------0	------00 00000000	
0x028	-	-	WTDIVEN[B,H,W]	WTDIV[B,H,W]
			------00	----0000
0x02C	-	-	-	WTCALPRD[B,H,W]
				--010011
0x030	-	-	-	WTCOSEL[B,H,W]
				-------0
0x034 - 0xFFC	-	-	-	-

1.23 低速 CR 预分频器

低速 CR 预分频器 基址 ：0x4003_C000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	-	LCR_PRSLD[B,H,W]
				--000000
0x000 - 0x0FC	-	-	-	-

1.24 外设时钟选通

外设时钟选通 基址 ：0x4003_C100

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	CKEN0[B,H,W]			
	---1---- -------1 -------- 11-11-11			
0x004	MRST0[B,H,W]			
	-------- -------0 -------- 00-00-00			
0x008 - 0x00C	-	-	-	-
0x010	CKEN1[B,H,W]			
	-------- -------- -------- ------11			
0x014	MRST1[B,H,W]			
	-------- -------- -------- ------00			
0x018 - 0x01C	-	-	-	-
0x020	CKEN2[B,H,W]			
	-------- -----1-- 111----- -------0			
0x024	MRST2[B,H,W]			
	-------- -----0-- 000----- -------0			
0x028 - 0x0FC	-	-	-	-

1.25 智能卡 I/F

智能卡 I/F ch.1　　基址 ：0x4003_C980

基址 + 地址	寄存器			
	+3	+2	+1	+0
0x00	-	-	GLOBALCONTROL1[H,W] -0001000 00000000	
0x04	-	-	STATUS[H,W] --000000 00000001	
0x08	-	-	PORTCONTROL[H,W] 0000--00 00-0-0-0	
0x0C	-	-	DATA[H,W] -------0 00000000	
0x10	-	-	CARDCLOCK [H,W] 00000000 00101000	
0x14	-	-	BAUDRATE[H,W] 0000001 01110100	
0x18	-	-	GUARDTIMER[H,W] -------- 00000000	
0x1C	-	-	IDLETIMER[H,W] 00000000 00000000	
0x20	-	-	GLOBALCONTROL2[H,W] -------- ----1-00	
0x24	-	-	DATA_FIFO[H,W] -------0 00000000	
0x28	-	-	FIFO_LEVEL_READ[H,W] 00000000 00000000	
0x2C	-	-	FIFO_LEVEL_WRITE[H,W] 00000000 00000000	
0x30	-	-	FIFO_MODE[H,W] 00000000 ----0000	
0x34	-	-	FIFO_CLEAR_MSB_WRITE[H,W] -------- -------0	
0x38	-	-	FIFO_CLEAR_MSB_READ[H,W] -------- -------0	
0x3C	-	-	-	-
0x40	-	-	IRQ_STATUS[H,W] -------- 00000000	
0x44-0x7C	-	-	-	-

1.26 MFSI2S

MFSI2S ch.4　　基址 ：0x4003_CA00

MFSI2S ch.6　　基址 ：0x4003_CA80

基址 + 地址	寄存器			
	+3	+2	+1	+0
0x00	-	-	CNTLREG[B, H,W] -----0-0 -0000-01	
0x04	-	-	I2SCLK[B, H,W] 00------ 00000000	
0x08	-	-	I2SST[B,H,W] ------00	I2SRST[B,H,W] 00000000
0x0C-0x3C	-	-	-	-

1.27 USB

USB ch.0 基址 ：0x4004_0000

基址+ 地址	寄存器			
	+3	+2	+1	+0
0x2100	-	-	HCNT1[B,H,W] -----001	HCNT0[B,H,W] 00000000
0x2104	-	-	HERR[B,H,W] 00000011	HIRQ[B,H,W] 0-000000
0x2108	-	-	HFCOMP[B,H,W] 00000000	HSTATE[B,H,W] --010010
0x210C	-	-	HRTIMER(1/0)[B,H,W] 0000000000000000	
0x2110	-	-	HADR[B,H,W] -0000000	HRTIMER(2)[B,H,W] ------00
0x2114	-	-	HEOF(1/0)[B,H,W] --00000000000000	
0x2118	-	-	HFRAME(1/0)[B,H,W] -----00000000000	
0x211C	-	-	-	HTOKEN[B,H,W] 00000000
0x2120	-	-	UDCC[B,H,W] -------- 10100-00	
0x2124	-	-	EP0C[H,W] ------0- -1000000	
0x2128	-	-	EP1C[H,W] 01100001 00000000	
0x212C	-	-	EP2C[H,W] 0110000- -1000000	
0x2130	-	-	EP3C[H,W] 0110000- -1000000	
0x2134	-	-	EP4C[H,W] 0110000- -1000000	
0x2138	-	-	EP5C[H,W] 0110000- -1000000	
0x213C	-	-	TMSP[H,W] -----000 00000000	
0x2140	-	-	UDCIE[B,H,W] --000000	UDCS[B,H,W] --000000
0x2144	-	-	EP0IS[H,W] 10---1-- --------	
0x2148	-	-	EP0OS[H,W] 100--00- -XXXXXXX	

基址+ 地址	寄存器			
	+3	+2	+1	+0
0x214C	-	-	EP1S[H,W] 100-000X XXXXXXXX	
0x2150	-	-	EP2S[H,W] 100-000- -XXXXXXX	
0x2154	-	-	EP3S[H,W] 100-000- -XXXXXXX	
0x2158	-	-	EP4S[H,W] 100-000- -XXXXXXX	
0x215C	-	-	EP5S[H,W] 100-000- -XXXXXXX	
0x2160	-	-	EP0DTH[B,H,W] XXXXXXXX	EP0DTL[B,H,W] XXXXXXXX
0x2164	-	-	EP1DTH[B,H,W] XXXXXXXX	EP1DTL[B,H,W] XXXXXXXX
0x2168	-	-	EP2DTH[B,H,W] XXXXXXXX	EP2DTL[B,H,W] XXXXXXXX
0x216C	-	-	EP3DTH[B,H,W] XXXXXXXX	EP3DTL[B,H,W] XXXXXXXX
0x2170	-	-	EP4DTH[B,H,W] XXXXXXXX	EP4DTL[B,H,W] XXXXXXXX
0x2174	-	-	EP5DTH[B,H,W] XXXXXXXX	EP5DTL[B,H,W] XXXXXXXX
0x2178 - 0x217C	-	-	-	-

1.28 DSTC

DSTC 基址 ：0x4006_1000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x0000	DESTP[B,H,W] 00000000 00000000 00000000 00000000			
0x0004	HWDESP[B,H,W] 00XXXXXX XXXXXX00 00000000 00000000			
0x0008	SWTR[H] 00000000 00000000		CFG[B] 01000000	CMD[B] 00000001
0x000C	MONERS[B,H,W] 00XXXXXX XXXXXX00 XXXXXXXX XXX00000			
0x0010	DREQENB[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x0014	DREQENB[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x0030	HWINT[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x0034	HWINT[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x0050	HWINTCLR[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x0054	HWINTCLR[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x070	DQMSK[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x074	DQMSK[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x090	DQMSKCLR[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x094	DQMSKCLR[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x0098 - 0x0FFC	-	-	-	-

1.29 MTB_DWT

MTB_DWT 基址 ：0xF000_1000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	CMP_ADDR_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x004	CMP_DATA_START[B,H,W]			
	00000000 00000000 00000000 00000000			
0x008	CMP_MASK_START[B,H,W] 00000000 00000000 00000000 00000000			
0x00C	-	-	-	-
0x010	CMP_ADDR_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x014	CMP_DATA_STOP[B,H,W]			
	00000000 00000000 00000000 00000000			
0x018	CMP_MASK_STOP[B,H,W] 00000000 00000000 00000000 00000000			
0x01C	-	-	-	-
0x020	-	-	-	FCT[B,H,W]
				00000000
0x024 - 0xFCC	-	-	-	-
0xFD0	PID4[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0xFD4	PID5[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0xFD8	PID6[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0xFDC	PID7[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0xFE0	PID0[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0xFE4	PID1[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0xFE8	PID2[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0xFEC	PID3[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0xFF0	CID0[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0xFF4	CID1[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0xFF8	CID2[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0xFFC	CID3[B,H,W] XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			

1.30 快速 GPIO

快速 GPIO 基址 ：0xF800_0000

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	-	-	FPDIR0[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x004	-	-	FPDIR1[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x008	-	-	FPDIR2[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x00C	-	-	FPDIR3[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x010	-	-	FPDIR4[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x014	-	-	FPDIR5[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x018	-	-	FPDIR6[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x01C	-	-	-	-
0x020	-	-	FPDIR8[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x024 - 0x034	-	-	-	-
0x038	-	-	FPDIRE[B,H,W]	
			XXXXXXXX XXXXXXXX	
0x03C	-	-	-	-
0x040	-	-	FPDOR0[B,H,W]	
			00000000 00000000	
0x044	-	-	FPDOR1[B,H,W]	
			00000000 00000000	
0x048	-	-	FPDOR2[B,H,W]	
			00000000 00000000	
0x04C	-	-	FPDOR3[B,H,W]	
			00000000 00000000	
0x050	-	-	FPDOR4[B,H,W]	
			00000000 00000000	
0x054	-	-	FPDOR5[B,H,W]	
			00000000 00000000	
0x058	-	-	FPDOR6[B,H,W]	
			00000000 00000000	
0x05C	-	-	-	-
0x060	-	-	FPDOR8[B,H,W]	
			00000000 00000000	
0x064 - 0x074	-	-	-	-

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x078	-	-	FPDORE[B,H,W]	
			00000000 00000000	
0x07C	-	-	-	-
0x080				M_FPDIR0[B,H,W]
				XXXXXXXX
0x084	-	-	-	M_FPDIR1[B,H,W]
				XXXXXXXX
0x088– 0x0BF	-			
0x0C0	-	-	-	M_FPDOR0[B,H,W]
				00000000
0x0C4	-	-	-	M_FPDOR1[B,H,W]
				00000000
0x0C8 - 0x0FC	-	-	-	-

1.31 VIR

VIR 基址 ：0xF800_0100

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x000	VIR00[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x004	VIR01[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x008	VIR02[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x00C	VIR03[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x010	VIR04[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x014	VIR05[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x018	VIR06[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x01C	VIR07[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x020	VIR08[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x024	VIR09[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x028	VIR10[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x02C	VIR11[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x030	VIR12[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x034	VIR13[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x038	VIR14[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x03C	VIR15[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x040	VIR16[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x044	VIR17[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x048	VIR18[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			

基址	寄存器			
+ 地址	+3	+2	+1	+0
0x04C	VIR19[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x050	VIR20[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x054	VIR21[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x058	VIR22[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x05C	VIR23[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x060	VIR24[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x064	VIR25[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x068	VIR26[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x06C	VIR27[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x070	VIR28[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x074	VIR29[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x078	VIR30[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			
0x07C	VIR31[W]			
	XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXX			

E.注意事项列表

本节说明各功能注意事项。

1. **高速 CR 用为主控时钟时的注意事项**

代码：9APRECAUTION-FM0-C03.0

1. 高速 CR 用为主控时钟时的注意事项

本节说明高速 CR 用为主控时钟时的注意事项。

高速 CR 的频率随温度和/或电源电压而变化。下表为高速 CR 用为主控时钟时各功能宏相关注意事项。另外，要注意高速 CR 用为 PLL 输入时钟以及选择 PLL 所用主控时钟的注意事项。

各宏注意事项

宏	功能/模式	注意事项
内部总线时钟	HCLK/FCLK/PCLK0/PCLK1	高速 CR 最高频率不得超过所用产品《数据手册》规定的内部操作时钟频率。
定时器	多功能定时器 基本定时器 监视定时器 双定时器 看门狗定时器 正交定时器	各宏定时器计数值应考虑高速 CR 频率变化。
A/D 转换器	采样时间 比较时间	考虑到高速 CR 的频率变化，A/D 转换器的采样时间以及比较时间应满足所用产品《数据手册》规定的规格。
USB, CAN	-	如果频率精确度不能达到规定要求，高速 CR 用为主控时钟时不能使用这些宏。
多功能串口	UART	因为高速 CR 的频率变化原因，波特率设置可能变得更差。 如果有波特率超出范围错误，则不能使用此功能/模式。
	CSIO, I2C, MFS-I2S	各宏的通讯应考虑高速 CR 频率的变化。
	LIN	如果不能满足规定的频率精确度，本功能不用于主机。 从机时，可以使用此功能。 如果是从机，高速时钟最高/最低频率时规定波特率会出现更多错误。因此，如果超过波特率错误限值，不能使用此功能。
智能卡接口	-	因为高速 CR 的频率变化原因，板速率设置可能变得更差。 如果有板速率超出范围错误，则不能使用此功能/模式。

Revision History

Document Revision History

Document Title: 32 位微处理器 FM0+ 家族 外设手册 Document Number: 002-11330			
Revision	**ECN No.**	**Origin of Change**	**Description of Change**
**	5285688	TEKA	New Specification (本文档版本号为 Rev**，译自英文版 002-04969 Rev **) Updated to Cypress format.